Gakken

合格情報処理
2月号第2付録
二〇〇〇年二月一日発行／第十六巻第二号
毎月一回一日発行

情報化の課題
産業社会と情報化
情報処理システム
ファイルとデータベース
システム開発とプログラミング
アルゴリズムとデータ構造
情報システム辞典

Printed In Japan

情報システム辞典
二〇〇〇年版
Gakken

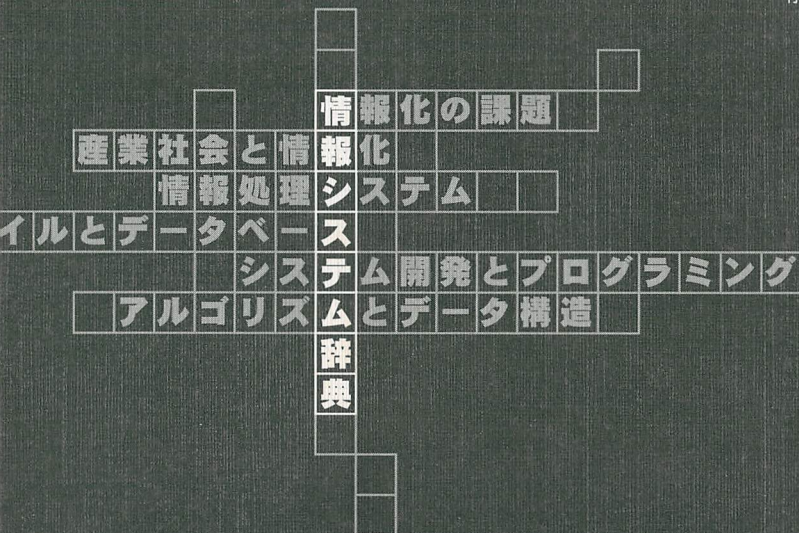
通産省情報処理技術者試験
2000年2月1日発行／第16巻第2号
毎月1回1日発行

情報システム辞典

システム開発とプログラミング
ファイルとデータベース
アルゴリズムとデータ構造
情報処理システム
産業社会と情報化／情報化の課題

二〇〇〇年版

学研 合格情報処理　2月号第2付録

通産省情報処理技術者試験対策

情報システム辞典 2000年版

アカサタナハマヤラワ英字

情報処理技術を学ぶうえでも、
語るうえでも
技術用語を正しく
理解しておくことは大切なことです。
情報処理用語は
簡単な語でも日常語とは異なり、
特定の意味で用いられたり、
特殊な使われ方をすることがあります。
これらを正しく知ることは、
勉強のためにも、将来の仕事のためにも
必要なことです。

学研

採択用語について

☆本辞典は、第二種・第一種共通カリキュラム、シスアドカリキュラムに示された、システム開発とプログラミング、ファイルとデータベース、アルゴリズムとデータ構造、情報処理システムの重要用語を中心に、その周辺の用語までを収録したものです。

☆「合格情報処理」別冊付録の「情報処理基本用語辞典」(ハード/ソフト/情報処理一般)、「パソコン・シスアド用語辞典」、「通信ネットワーク用語辞典」とともにご利用ください。重複を極力抑えましたので、併用していただくと便利に使えます。

用語の表記と掲載順

☆見出し語の表記、訳語は一定していない現状ですが、カリキュラムの表記、または共通テキストの表記に準じました。なお不十分と思われる語は、慣用の語や別の見出し語として収録しました。

☆用語は五十音順に配列しました。片仮名の長音は前の字の母音とみなします(つまり長音符号を無視しない)。国語辞典の標準的な配列法に準じました。

また、拗音、促音、濁音、半濁音は清音と見なします。

(例:「モジュール」は「もしゆうる」、「バーチャル」は「はあちやる」)

☆英語・英略語は混在させ、アルファベット順に配列しています。大文字と小文字は区別しません。

☆見出し語に続く( )は、その項目に解説があることを示します。解説末尾の( )は、その項目をあわせて参照すると理解が深められることを示します。

(同)=同義語、言い換え語　　(類)=類義語、ほぼ同義語
(対)=反対語、対照語、対語　　(参)=参照項目、参考項目
(関)=関連語、同族語　　(例)=例示項目

☆見出し語中の/や:は、言い換え、または略語であることを示します。

執　筆

鮎沢　正治	新井　忠明	飯島　秀之	池田　美保	板野　敏貴
稲場　久男	内田　義也	河内　栄一	川崎　清隆	川真田政夫
木幡　　悟	坂井　浩二	酒井　力郎	澤村　英仁	進藤　良雄
鈴木真由実	高澤　洋子	高橋　　勇	高橋　健吉	滝口　郁志
鶴岡　玄光	寺沢　康夫	豊住　昌雄	西村　修一	長谷川弘子
秦　　　浩	林　　　篤	増井　和也	真々田　稔	守　　章彦
山口　行広	山本　和明	脇山　　久		

■表紙デザイン/石井知哉　■DTP・図版/(株)メビウス

あいまい検索[ambiguous searching]

データベースや文書中の文字列データの検索を行うとき、文字コードの一致を前提としないで許容範囲を広げて文字列を検索すること。例えば、英字の大文字と小文字、全角文字と半角文字、ひらがなとカタカナの区別をしない検索をさす。

さらに許容範囲を広げた類義語検索、具体的な文字列を指定しないワイルドカード検索、文字の位置などが指定できる正規表現検索などがある。

(参)類義語検索、ワイルドカード

アクセス[access]

ユーザがデータベースやコンピュータシステム中のプログラムやファイルに到達すること、またそこに至る経路のこと。これにともなうコンピュータや通信装置の動作などを含む。一般に、次のようなプログラム処理やファイル操作、システム利用を単に「アクセス」「アクセスする」などという。

(1) プログラムが、主記憶装置や外部記憶装置などのデータを読み出したり、データを書き込んだりすること。

(2) ユーザがファイルやデータベースの情報を読み書きすること。

(3) 端末がネットワークを介して、ホストコンピュータのシステムを利用すること。

アクセス管理[access right management／access control]

共用可能な情報資源を管理、制御すること。具体的には、アクセスのログを残し、利用者のアクセス要求とアクセス権を検証し、不当なアクセスを制限し、警告を発するなどの処理を行う。複数のユーザが同時にファイルにアクセスすることで混乱や誤謬が生じないように管理したり、ユーザに課金する場合は課金情報を管理することも含む。アクセス制御はデータ破壊やシステムの不正利用を防ぐためだけでなく、障害の復旧、ユーザへの情報提供に重要な役割を担っている。

(関)排他制御、アクセス権

アクセス権[access right]

広義には、システムの共有資源を一時的に独占して利用する権利。利用権、参照権ともいう。端末利用権、CPU利用権、データ利用権、OS利用権、回線利用権などがある。特に利用者または作業工程に対して与えられたファイルの利用権をさすことがある。アクセス権を与えることをアクセス許可(access permission)という。これには、読取り許可、書込み許可、読取り書込み許可がある。(参)パスワード

ア

アクセス方式[access method]

プログラムによるデータやレコードの読込み、書出し、およびデータやレコードの取扱い方法。アクセス法ともいう。レコードの集合であるファイルは、記憶媒体に記録されるとき各種の編成方法(論理構造)が与えられるが、それぞれに応じて適切なアクセスを行うアクセス法がオペレーティングシステムにより提供される。順次アクセス(SAM)、ランダムアクセス(DAM)、索引順次アクセス(ISAM)、仮想記憶における索引順次アクセス(VSAM)などに分けることができる。

アサイン[assign]

ひとつのジョブを実行する際に必要となる装置やファイルを実行に先立って確保すること。ジョブは通常、磁気ディスク記憶装置、プリンタなどの周辺装置やさまざまなファイルを使用するが、これらはハード的に接続されてさえいればいつでも使用できるわけではなく、そのつど必要なものを確保する。汎用コンピュータのOSでは、通常ジョブ制御言語(JCL)により、装置名、ボリューム名、ファイル名などを指定してこの処理を行う。(関)JCL、(参)アロケーション

アスキーコード[ASCII code] (同)ASCII

アスキーセーブ[ASCII code save] (関)セーブ

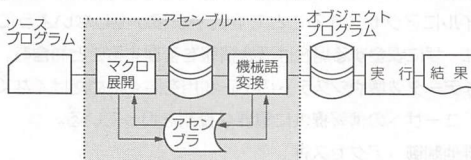
アセンブラ[assembler]

アセンブラ言語で書かれたソースプログラムを機械語のオブジェクトプログラムに翻訳する言語処理プログラム。アセンブラの命令語には、

(1)実行時の命令コードと1対1に対応する命令語

(2)アセンブル過程に指示を与えるアセンブリ命令

(3)アセンブル過程で一連の機械語に置き換えるマクロ命令などがある。

(参)マクロアセンブラ、アセンブラ言語

アセンブラ言語[assembly language]

2進数による機械語の命令を書くわずらわしさを解消するために作られた言語。例えば、加算はA、ロードはLというように表意コードを用いて書き表す。記号言語ともいう。高水準言語と異なり、コンピュータの機種に依存した記述になる。コンピュータ資源を効率よく使用するプログラムが組める。プログラムのステップを短くできる。しかし、機械に依存した言語であり、ハードウェアを熟知していないと効率のよいプログラミング

ができない。またプログラミングの時間もコンパイラ言語を用いるより多く必要とするなどの難点があって、使用する局面は少なくなっている。ハードウェアを直接制御する場合や制御系のプログラミングに使用することが多い。(類)機械向き言語

アセンブル[assemble]

アセンブラ言語で書かれたソースプログラムをアセンブラによりオブジェクトプログラム(機械語コード)に翻訳する作業。

(1)**セルフアセンブル**:実行用の計算機で、ソースプログラムをアセンブルすること。通常のアセンブル法で、自機種用のアセンブラがある機種で可能。クロスアセンブルと特に区別するときの用語。

(2)**クロスアセンブル**:ある計算機用のプログラムを、それとは異なるアーキテクチャをもつ別の計算機上でアセンブルすること。この結果、ある計算機上で働く機械語を別の計算機上で生成できる。生成したオブジェクトプログラムを実行するコンピュータをオブジェクトコンピュータ、またはターゲットコンピュータという。実行用コンピュータがプログラム開発に自由に使用できないとき、実行用コンピュータを使用するより開発の効率が良い場合などに行う。

値呼出し[call by value]

手続きや関数を呼び出して使用するとき、パラメタの引き渡しを仮引数と実引数の対応で行うこと。この場合、値を渡された手続きや関数の中で値を書き換えても、呼び出した側の値に影響はない。Cでは値渡し、COBOLでは値引渡しという。(関)引数、(対)アドレス渡し

圧縮[compression/compaction] (参)データ圧縮

アップグレード[upgrade]

ハードウェアやソフトウェアを増強したり拡張したりすること。強力なマイクロプロセッサや周辺機器を加えたり、新しいプログラムを入れることで、システムを高機能化、高性能すること。特に、全部のハードウェアを入れ換えるときはリプレース、一部のファイルの置き換えについてはアップデート、プログラムを新しいバージョンに代えることはバージョンアップという。

アップデート[update]

プログラムやファイルを最新のものに置き換えること。(同)更新

後入れ先出し方式[last in first out:LIFO] (同)LIFO

アトリビュート[attribute]

コンピュータで扱うデータやファイルがもつ性質のこと。属性ともいう。例えば、次のような性質を属性という。

(1)ディスプレイに表示するデータは、対応する属性によって高輝度、反

転、明滅などをする。印字する字体や大きさ、表示色なども属性である。画像処理では、図形の解像度、座標の表し方、色などの情報をさす。

(2) レコードの場合には、レコード長、レコード形式、データセット名、ボリューム識別番号、用途、作成日などが属性である。

(3) ファイルアクセスの場合には、読出し専用、システム専用、秘匿などの性格づけが属性となる。

(4) データベースの関係モデルを表現した二次元表の列のこと。このとき行は組(tuple)という。(参)関係データベース、(対)組

(5) データモデルのひとつであるE-Rモデルで、実体と結び付く(関連づける)値を定義する性質。例えば、ある人物A(実体)を記述するデータは無数にあるが、その中から(性別、生年月日、身長、体重)という属性に注目すると、人物Aに関連づけた値(実現値)は(男、19741103、171、63)などとなる。(関)E-Rモデル

アドレス[address]

広義には情報の読込み先、転送先を指定するための識別文字列。ある場所と別の場所を区別するためにシステムが付けた地名、番地などの名前の集まり。広く次のような名前をさす。

①レジスタ、主記憶装置、補助記憶装置などの中の記憶場所を示すための識別文字の集まり。番地ともいう。記憶装置を多数の電子的な記憶単位の集合として扱い、各単位を識別するために一連の番号(絶対アドレス)を付けておき、読み取ったり書き込んだりする際にはこの番地を使って記憶装置内のプログラムやデータを指定する。主記憶装置内では、通常 1ワード分が 1番地分になる。

(関)実効アドレス、相対アドレス

②ソースプログラム内では変数や構造体などのオブジェクトのメモリ上の位置を示す。プログラム言語によっては絶対アドレスのほか、ラベル名、記号アドレスで表現することもできる。

アドレスカウンタ[address counter] (同)命令アドレスレジスタ

アドレス指定[addressing]

①コンピュータの機械語レベルで、命令をどのデータに作用(オペレート)させるか決定すること。機械語の命令は、コンピュータにデータ処理の種類を指定する命令部と処理対象のデータの所在を指定するアドレス部とからなっている。アドレス指定の方式には、(1)直接アドレス指定、(2)間接アドレス指定、(3)即値アドレス指定などがある。

②ファイル編成の分野では、直接編成ファイルのアドレス生成の意味でアドレス指定という語を使うことがある。

アドレス修飾[address modification]

アドレス指定の方式のひとつ。命令語のアドレス部の内容を一定の規則のもとに変更し命令の作用対象となる絶対アドレスを決定すること。命令自体を変えずに処理対象となるデータのあるアドレスを変えることができるので、記憶装置の融通性のある利用が可能になる。

次のような方式がある。

(1)インデックスレジスタ修飾(インデックスアドレス指定)

(2)ベースレジスタ修飾(ベースアドレス指定)

(3)相対アドレス方式(相対アドレス指定)などがある。

アドレス生成[address generation]

直接編成ファイルのレコードを記憶装置に書き込む物理アドレスを算出すること。直接編成ファイルではあるレコードは、他のレコードと区別できるキー項目を備えている。この項目の値から直接アクセス記憶装置(DASD)上の相対レコード番号を求め、その位置にレコードを記録する。相対レコード番号とは、ファイルの先頭のレコードを0として、先頭から数えた番号のことである。実際の入出力は、OSのデータ管理機能により相対レコード番号を直接アクセス記憶装置の物理アドレスに変換し、その位置にレコードを記憶する。このような手順でアドレスを作りだすことをアドレス生成という。(1)直接アドレス方式と(2)間接アドレス方式がある。(関)直接アドレス方式、間接アドレス方式

アドレス部[address part]

①コンピュータの命令語の構成要素で、命令を働きかけるアドレスを決定するための部分。(関)命令

②ファイル編成の分野では、カウント部をさすことがある。(同)カウント部

アドレス変換[address translation]

①仮想記憶方式において、仮想アドレスで示されているアドレスを実記憶装置のアドレスに変換すること。実際にはプログラムの実行時に先だち、まず仮想アドレスを作り、これをセグメント番号やページ番号、変位を考慮のうえ、実行時にはじめて実記憶空間のアドレスに変換する。この変換を行う機構を動的アドレス変換機構(DAT)という。

(参)仮想記憶、セグメント

②ファイル編成の分野では、アドレス生成の意味で、アドレス変換、アドレス変更という語を使うことがある。

アドレスレジスタ[address register]

CPUの制御部分にあるレジスタ。命令実行段階において頻繁に用いる主記憶アドレス情報を一時的に保存しておくためのレジスタ。メモリアドレスレジスタともいう。

アドレス渡し[call by reference／call by address]

手続きや関数を呼び出して使用するとき、パラメタの引き渡しを仮引数のアドレスやポインタで与えて、実引数との対応をとること。仮引数で値を変更した結果が実引数に反映されるので、値の変更結果を呼出し元の関数側で参照できる。(対)値呼出し、(関)引数

アプリケーションソフト[application software]

業務で扱うデータを処理するプログラム。システムプログラム(OS)、ミドルウェア(サービスプログラムやプログラム言語など)、共通ソフトウェア(日本語FEPや通信ソフトウェアなど)に対比する用語。応用プログラム、あるいは適用業務プログラムともいう。

ユーザが自分の業務のために作成するプログラム、あるいはソフトウェアハウスに委託して作成するプログラムである。具体的には、在庫管理、工程管理、販売予測など特定の目的でデータ処理するプログラムをさす。パソコンの場合、コンピュータメーカやソフトウェアハウスが作成し、マニュアル(取扱い説明書)とともに販売している。このように既製品化したプログラムのことを、特にアプリケーションパッケージ、または略してアプリケーションという。

(参)カスタマイズ、ソフトウェアパッケージ

あふれ[overflow]　(同)オーバフロー

あふれ域[overflow area]　(関)索引順次編成

あふれ検査[overflow check]　(同)オーバフロー検査

アベンド[abend／abnormal end of task]

タスクの異常終了。タスクがエラーを起こして処理を続けられなくなり終了すること。または、他のタスクやジョブ等に悪影響を与えるため、制御プログラムによって異常終了させられること。(類)アボート

アボート[abort／abnormal termination]

コンピュータで実行中の処理が、ハードウェアの異常やプログラムのエラーにより、続行が不可能、または望ましくない状態になったとき、OSまたはオペレータにより処理を中断すること。異常終了、停止ともいう。プログラムが異常終了すると、一般的なプログラムの後処理(オープン中のファイルのクローズ、出力中のデータのフラッシュ等)は行われない。

誤り回復[error recovery]

ハードウェアやソフトウェアに誤りが発生したとき、所定の状態に復帰させるため、誤りを解析し、回復させること。例えば、中断されたジョブを再試行したり、伝送中にエラーが発生したデータの再送要求を行うことをいう。(参)障害回復

アルゴリズム[algorithm]

与えられた問題を機械的に解くために組み立てる、あいまいさのない解法。算法ともいう。アルゴリズムは、順序、分岐、繰返しのある有限個の操作(命令の組合せ)で定義してあり、文や流れ図でも表現できる。計算機で解くためにプログラム言語で表現するとプログラムになる。プログラムは、有限回の適用(必ず実行が終了する)によって所定の解答を得る。広義には処理手順(手続き)一般をさす。(類)段階的詳細化

アルファニューメリック[alphanumeric]

英数字。これに制御文字、特殊文字、間隔文字(空白文字)を加えた1バイト(半角)の文字集合をさすこともある。事実上、ソースプログラムを記述するときに使用できる文字である。(関)ASCII、EBCDIC

アルファニューメリックチェック[alphanumeric check]

入力データのチェックの一手法。データは英数字と規定した項目に、誤って英数字以外の文字が混入するのを防ぐために行う。

アロケーション[allocation]

プログラムを実行する際、処理の必要に応じてファイルや記憶領域を割り当てること。マルチプログラミングでは、いつ、どの番地にプログラムを格納するか、実行時にならないと決まらないので、プログラムは相対番地で作成する。OSは実行時に主記憶上の空き領域内にプログラムを格納する。(参)アサイン

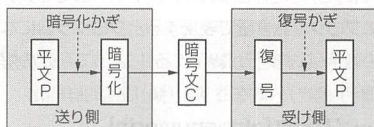
アンインストール[uninstall]

使用中のハードディスク装置から不要になったソフトやデータを削除するとともに、初期化ファイルやショートカット、メニューなどの内容をインストール前の状態に戻すこと。アンインストーラというソフトにより処理する。GUIのソフトウェアは多数のファイルから構成されており、使用のつどシステム用のファイルが書き換えられていることもある。手作業でインストール前の状態に戻すのは容易ではなく、この作業の自動化機能を有するアンインストーラを使用して行うのが通例である。
(対)インストール

暗号[cryptography／cipher／cypher]

データの内容を第三者が判読できないように組織的に変換する方法、または変換したデータ。情報セキュリティのための重要な技術のひとつである。暗号データにすることを暗号化と呼び、暗号データを元のデータに戻すことを復号と呼ぶ。個人のプライバシに関わるデータ、競争相手に知られると自分の優位性が保てなくなるようなデータ、有料データなどについて行われる。通信回線(特に無線)は傍受が容易なため、機密度の高いデータは暗号化して送信する。送信側は暗号かぎによって暗号化し、受信側は

復号かぎで復号する。(対)復号

【暗号化かぎ】 情報を暗号化および復号する際、それぞれの変換に使用するパラメタ。暗号化かぎ(encryption key)と復号かぎ(decryption key)がある。

【暗号化方式】 暗号化方式には大きく二つの系統がある。

(1)**慣用暗号系**(秘密かぎ暗号方式):暗号化キーと復号キーが同一の方式(キーの秘密性が重要な課題)(同)慣用暗号系、(例)DES

(2)**公開かぎ暗号系**:暗号化かぎと復号かぎが異なり、前者を公開、後者を秘密にする方式。(同)公開かぎ暗号系、(例)RSA

安全性[security] (類)セキュリティ

安全対策基準 (同)情報システム安全対策基準

アンダフロー[underflow]

下位桁あふれ。演算を行ったとき、その結果が処理装置のレジスタが取り扱える最小値(絶対値で)よりも小さくなった状態。浮動小数点の演算において、許された範囲に入らない負の指数が発生したときに、アンダフローとなる。アンダフローになると内部割込みが発生する。(対)オーバフロー

アンチウィルスプログラム[anti-virus program]

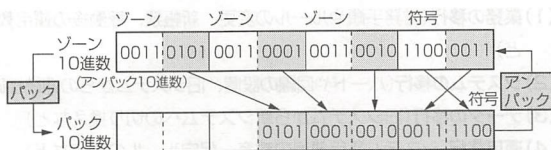
システムのメモリやディスクに侵入しているコンピュータウィルスを検査し、もし感染していたならばウィルスを除去するとともに、損害を受けたファイルなどの復旧作業を支援するユーティリティプログラム。知られているウィルスのコードを検出したり、ファイルのチェックサムを計算して、前回記憶したチェックサムと照合したりして、異状を検出する。一般にはワクチンとも呼ぶ。

(参)コンピュータウィルス

安定な整列アルゴリズム[stable sort algorithm]

整列アルゴリズムの呼称で、同じキー値をもつデータが二つ以上あったとき、それらのデータは、整列後も整列前の順序の通りに並べるアルゴリズム。例えば、氏名の五十音順に並んだ名簿ファイルを生年月日で並びかえたとき、生年月日の同じ人が 6人いたとすれば、その 6人の並びの部分はやはり氏名の五十音順に整列させるアルゴリズム。具体的にはビンソート(bin sort)というアルゴリズムがある。

アンパック[unpack]

パック形式の10進数値をゾーン形式に変換すること。(対)パック

アンパック10進数[unpack format]

コンピュータ内の数値の表し方のひとつ。1バイトで1桁の数字を表し、小数点の位置をLSBの右と想定し、符号は最下位バイトの上位4ビットにおく。アンパック形式、ゾーン形式ともいう。

(同)ゾーン10進数、(対)パック10進数

アンバンドリング[unbundling]

ハードウェアとソフトウェアの価格を分離して販売すること。コンピュータのハードウェア、ソフトウェア、教育、保守サービスなどそれぞれを別建ての価格で販売する方式。両者をまとめて提供するバンドリングは、主にパソコンの分野で、初心者向けにインストール作業の省略、不法な複製を防止する方法として行われている。

暗黙の型宣言[default declaration／implicit typing]

プログラムの中にデータの型や長さなどの属性を明示的に指定しなかったとき、システム(コンパイラ等)が自動的にデータの型や長さを選択すること。例えば、Fortranの場合、何も宣言しないときには、変数名の最初の1文字で定める。最初の文字がIからNの間にあれば整数型と定義し、それ以外の文字の場合は実数型と定義する。暗示的宣言ともいう。

(参)型、デフォルト

暗黙の型変換[implicit type conversion]

型変換の関数を使わないで行われる型変換。計算式の代入の際に異なる型の変数を用いると、処理系の解釈で作用対象の型を変更する。

アンロード[unload]

仮想記憶上、または主記憶上に常駐しているファイルが更新処理を受けたときに、その内容を外部の記憶媒体上に書き戻すこと。

(対)ロード

イ

移行[systems conversion]

従来の業務システムを新規の業務システムに変更、切り換えする工程を広く移行という。移行は、次の4つの領域で考える。

(1)**業務の移行**(業務手順やルールの変更、新帳票、稼働後の運用教育など)
(2)**システムの移行**(ハードや回線の設置、旧システムからの連続性確保)
(3)**データの移行**(旧システムから新システムへの切り換えなど)
(4)**運用移行**(システム運用要員の教育、保守ルールの確定など)

移植[port／porting]

あるハードウェアやOSのもとで利用できるソフトウェアを手直しして、別のハードウェアやOSで利用できるように修正すること。ソフトウェア開発の効率を高めるために、既存ソフトウェアを活用しようという考え方から生まれた。ソースプログラムをコンパイルし直せばそのまま別機種で実行できる場合、データ媒体、データが他機種で利用できる場合は、「互換性がある」という。
(参)互換性

移植性[portability]

プログラムの移植作業のしやすさ。移植性を高めるためにはもとになるソフトウェアの作成段階から次のような配慮が必要である。
(1)標準的な高水準言語を使用する
(2)特定の機種に依存する機能を、限られた数のモジュールに限定する

一意[uniqueness／identify]

プログラム中で、あるデータが他のデータと区別でき、特定のデータを指定できること。ファイル名、項目名、キー値、修飾語、添字などの綴りや扱いによって唯一のデータを指定すること。プログラム言語の種類や指定の方式により、ある名前の手続きや変数が一意に定まる範囲は異なる。

一時ファイル[temporary file／transaction file]

永久ファイル(マスタファイル)に対比した語で、短期間使用するファイル。(例)トランザクションファイル、テンポラリファイル

1バイト文字コード[one-byte character code]

(同)ANK、(関)2バイト文字コード

一括コンパイル[one-pass compile]　(参)コンパイラ

一括処理方式[batch processing system]　(同)バッチ処理

一括翻訳[one-pass compile]　(参)コンパイラ

一斉テスト[whole test]

小規模なシステムのテスト法で、モジュールテスト(単体テスト)や結合テストを省略して、各モジュールの開発が終わった時点ですべてのモジュールを結合して行うテスト。エラーが存在した場合、原因の特定が困難な手法で、大規模なシステムのテストには適していない。

イニシャライズ[initialize]

システム利用に先だって行う処理一般。(同)初期化

イニシャルプログラムローダ[initial program loader：IPL]

(関)制御プログラム

イベント駆動[event driven]

イベントが発生した時点で、対応する処理が実行されるようにプログラム内の制御を書いておくこと。イベントとは、プログラム実行中に生ずるマウスのクリックやキー入力など、外部からの介入のこと。対話型のOSやアプリケーションでは、処理の流れは定型化されておらず、操作中のユーザの要求次第でシステムの対応を変える必要がある。イベント単位で独自に動作する多数の小プログラムを作っておき、システム全体として矛盾なく、スムーズに動作するアプリケーションプログラムの作成方法をイベント駆動型プログラミングという。

意味解析[semantic analysis]

翻訳プログラムの処理は、字句解析と構文解析を行い、その後、意味解析を行う。例えば型の概念のある言語では、変数が整数型、実数型などの型をもつが、各変数がもっている型の意味を解析し、変数名と型の一覧表を作成することなどがこれに相当する。

(参)構文解析、字句解析、コンパイル

イメージデータ[image data]

画像データのこと。普通、イメージデータは画像をごく小さな領域(ピクセル)に細分化し、白黒の場合そこが白か黒かのいずれか近いほうのデータにする。また、カラーの場合は、その領域の色に最も近い色素の組合せのデータになる。細分化の度合いを細かくすればするほど、実際の画像に近く見える。この細分化の度合いを画像の解像度、または画像品質と呼び、dpi(dot per inch)の単位で表す。(対)テキスト、(参)JPEG

イメージプロセッシング[image information processing]

画像、映像を構成する情報を処理すること。画像、映像情報の入出力・加工・処理だけでなく、画像の認識等も含む。医学用のCTスキャンやランドサット、ノア等の人工衛星からの地理情報処理などが実用化されている。画像情報処理ともいう。

入口点[entry／entry point]

プログラム中の特定の場所で、他のプログラムやルーチンから実行の制御が渡される場所のこと。例えばサブプログラムの先頭、サブルーチンの最初のアドレスなどが入口点になる。プログラム中に複数の入口点がある場合もある。入口点に対応する語が出口(exit)または出口点(exit point)で、実行中のプログラムやルーチンから抜け出る場所をさし、その命令の

実行後は制御が他の場所に移る。

入れ子[nest／nesting]

ある種の構造を、同じ種類の構造の中にひとつまたはいくつか組み入れること。入れ子の内部の処理が終了すると、ひとつ外側へ制御が戻る。プログラム言語の処理系により、入れ子の記法、限界値が定められている。入れ子の構造には次のような例がある。

(1) 再帰的なアルゴリズムは入れ子の構造をもっている。またプログラミングでも再帰的な構造を用いることは多い。(関)再帰
(2) 繰返し処理の中に同一形式の繰返し処理が入ること。(関)繰返し
(3) サブルーチンがサブルーチンを呼ぶこと。関数が関数を呼ぶこと。
(4) if文の構造の中に、if文が入ること。
(5) 割込み処理プログラムが、さらに優先度の高い割込み要因による割込みを受け、その処理プログラムを起動すること。

インクリメンタルサーチ[incremental search]

読みの順に並んだ文字列のテーブルをキーワードで探索する手法。キーワードのすべての文字が入力されてから探索するのではなく、キーワードの一文字入力のつど、探索範囲をせばめ、頭出しをする方式。辞典ソフトなどの見出し語検索に応用する。

インクリメント[increment／incremental／incrementation parameter]

①増分のこと。一定回数の繰り返しなどを制御する文(FortranのDO文、BASICのFOR…NEXT文など)では、繰り返すたびに回数を制御する変数やポインタに加算する一定の値。きざみともいう。増分の値が負の場合、特に減分という。(対)デクリメント

②レジスタや特定の記憶領域の値に一定の値を加えること、またはその加える値。カウンタやスタックポインタなどに用いる。

インサート[insert] (同)挿入

印字コード[print code]

コンピュータからプリンタに送られる文字出力のための情報。通常に印字される文字の情報交換用コードのほか、プリンタを制御する情報も含む。

印字速度[printing speed]

印刷装置が印字可能な単位時間当たりの文字数。通常は1秒当たりの文字数(CPS)で表すが、ページプリンタではより実際的な、1分当たりの平均出力ページ数(PPM)で表すことが多い。

インストール[install]

システムプログラムやアプリケーションプログラムを記憶装置に格納し

たり、環境を設定して、そのプログラムが動作可能にすること。プログラムの一部分を変更したり、実行のために他のプログラムを導入するような場合も含む。パソコンのアプリケーションプログラムは、フロッピーディスクやCD-ROMで提供され、そのままの状態では使用できない場合が多い。インストールに際してはディレクトリの作成や圧縮プログラムの展開、プログラムのコピー、環境設定が必要である。これらの作業の負担を軽減する自動処理プログラムにセットアッププログラム、インストーラ(installer／install program)などがある。

インストラクション[instruction]　(同)命令

インストラクションミックス[instruction mix]

コンピュータの命令の実行速度を評価する方法。命令ミックスともいう。コンピュータの処理能力を評価するために用いる。命令ごとの実行時間にそれぞれの重み(ウェイト)を掛けた数値の和で表す。和の値が小さい程内部処理能力が高い。それぞれの命令は、使用する頻度によって重み(ウェイト)を付ける。例えば、科学技術計算を意識してウェイト付けを行ったギブソンミックスがある。(参)ギブソンミックス

インスペクション[inspection]

もとは品質管理の用語で、製品を検査し、不良品の判定を下すこと。システム開発では、レビューの場での批評をさす。特に、後工程でソフトウェアに欠陥が発見された場合、前工程のどの段階で織り込まれたか調査し、その部分の担当者に説明させ、担当者の誤解や思い違いを明らかにすること。再度の誤りを防止することのほか、技量の向上を図るねらいもある。ソフトウェアの品質向上の手法であると同時にOJTによるエンジニアの教育の場でもある。(関)レビュー

インターネット[INTERNET／Internet]

世界各国の複数のコンピュータネットワークを相互に接続した、国際規模のコンピュータ通信ネットワーク。研究ネットワークARPANETに端を発し、米国の民間研究団体などのネットワークを吸収し巨大化してきた。

標準プロトコルは、ARPANETから生まれたTCP／IP。日本でも研究機関を中心としたいくつかの国内ネットワークがインターネットと接続していたが、インターネットの商用利用が解禁され、1993年秋から付加価値通信網(VAN)会社が、1994年からはパソコン通信会社がインターネットへの接続サービスを開始し、また便利なブラウザやメーラが普及したため、急速に利用ユーザが増えている。

(関)WWW、Web

インタフェース[interface]

①広義には、独立した二つ以上のシステムを接続し、効率よく動作させる

ために種々の調整を行うサブシステム。非常に幅広く使用される用語で「仲介」、「橋渡し」などに相当する。人間とコンピュータのインタフェースをヒューマンインタフェース、コンピュータシステム間のインタフェースを通信インタフェース、コンピュータとネットワーク間のインタフェースをユーザ・網インタフェース、ネットワークの交換機間のインタフェースを局間インタフェースなどと呼ぶ。

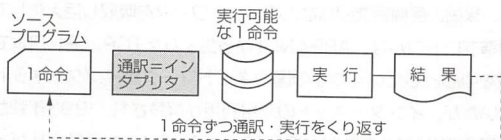
②狭義には、装置間でのデータのやりとりを正確に行うハードウェアやソフトウェア、またはその規格をさす。

(1)二つの装置を連結するハードウェア構成要素。二つ以上のプログラムによって共用する主記憶装置の一部、またはレジスタ。

(2)ソフトウェア間の規約。モジュール化したプログラムにおいてプログラム間インタフェース、モジュール間インタフェースとは、あるプログラムが別のプログラムを呼び出して利用する際に、データやパラメタを正しく授受し、他のプログラムやデータ領域に副作用(悪影響)を及ぼさないように定義することである。

(参)モジュール間インタフェース

インタプリタ[interpreter]

プログラム言語で記述したソースプログラムの命令、または中間コードを1命令ごとに機械語に翻訳しながら実行するプログラム。ソースプログラムをいちどに翻訳するコンパイラに対比して、解釈プログラム、通訳プログラムともいう。一般に、ひとつの命令に対して複数の機械語の命令をサブルーチンとして対応させている。インタプリタは少ない記憶容量でも動作するが、コンパイルされたプログラムに比べ処理速度が遅い。反面、未完成でも実行できる、実行しながらエラーを発見しその場で直せるというデバッグ上の利点がある。初期のパソコンが装備していたBASICは、ほとんどがインタプリタ形式であった。Javaアプレット(中間コード)を実行するソフトも広義のインタプリタといえる。

インタラクティブ[interactive]

対話型、会話型。キーボードやマウス、ディスプレイ等を介して人間とシステムが情報を交換しながら情報処理を進める方式。典型的な例としては、ワープロ操作、パソコンの表計算ソフトの操作がある。

機能の選択、データの入力、結果の出力など、人間の判断と操作を加え

ながら逐次に処理が進展する点で、あらかじめ処理手順を定めデータファイルを用意するバッチ処理とは異なる。なお、この用語は広く、ネットワークで結ばれたコンピュータ間の情報交換、コンピュータの制御下にある機械装置とコンピュータの情報交換に使用する場合もある。

インタリーブ[interleave/address interleave]

主記憶装置へのアクセス時間を短くする方式のひとつ。主記憶装置を複数のバンクに分けて使用し、CPUは複数のバンクに接続し、隣合うアドレスを別々のバンクに割り当てておく方式。先回り制御やマルチプロセッシングに向いている。

インタロック[interlock]

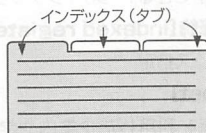
複数の要求に対して同時に応えられない装置で、ひとつの動作中に別の箇所から動作の要求があった時に、後からの要求には応じないようにすること、またはその機構。例えば、複数のタスクがメモリ等の共用資源を使用する場合に、インタロックはタスクの優先順位にしたがって、一時にひとつの要求しか受け付けないようにする。(参)排他制御

インデックス[index]

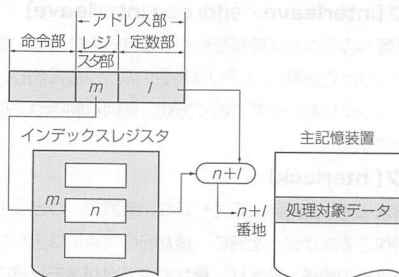
データの所在する位置を示す数字(例えばアドレス)について、別のデータを関連させてこれを変更(修飾)するための値。指標、修飾子ともいう。

(1) インデックスアドレス指定では、オペランドの示す値にインデックスレジスタの保持する値を加算して、データのある主記憶のアドレスを示す。インデックスレジスタの保持している値をインデックスという。

(2) ファイル中の特定のレコードや文書中の特定の項目の所在を早く知るためのキー。索引ともいう。一般にはキーと参照データのテーブルの形式になっている特定のデータにアクセスするときは、まずインデックスを探索し、そこから目的とするデータの位置を得てデータをアクセスする。(参)索引順次編成

(3) Windowsでは、フォルダの上部にはみ出して、標題などを書き込む領域。アイコンのカード、ノートなどに貼る「つまみ」に類似する。タブともいう。

インデックスアドレス指定[indexed addressing]

命令の処理の対象となるアドレスを指定する手法のひとつ。命令のアドレス部は、(1)基準アドレスを保持するインデックス(指標)レジスタ番号

を指定するレジスタ部、(2)そこからの変位を指定する定数部のふたつからなる。処理対象となるデータの番地を(m番レジスタの内容n＋アドレス部の数部 l)で指定する方式である。この方式を指標アドレス方式、インデックスレジスタ修飾ともいう。(参)アドレス指定

インデックスコンポーネント[index component]

索引は高速な検索を行うため階層構造をもつ。最下層の索引として、レコードを記録した各CI(制御インタバル)の情報を管理する順序セットが作られる。順序セットはそれぞれのCIの最後のレコードのキー値が記録されている。順序セットもまた、CIに記録されるが、この順序セットの索引CIの最後のキー値を記録する。これが索引セットである。索引セットはKSDSのレコードが多くなった場合は、索引セットのCIの最後のキーを記録する索引セットを作成する。このようにレコード数が膨大になった場合は複数の階層をもつことがある。

(関)制御インタバル、制御エリア

インデックス領域[indexed area] (関)索引順次編成

インデックスレジスタ[index register]

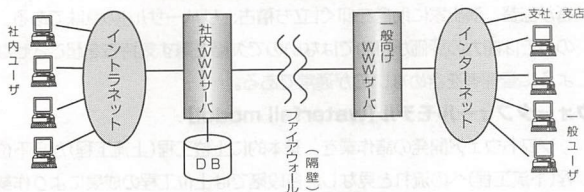
中央処理装置内のレジスタの一種。主としてアドレス指定を行い、ときにはカウンタとして使用するレジスタ。指標レジスタともいう。命令のアドレス部の定数にインデックスレジスタの内容を加え実効アドレスとする。アドレス指定に用いない場合は、レジスタの値に加算、減算を行うことでカウンタとして使用できる。(関)インデックスアドレス指定

インデックスレジスタ修飾[indexed register modification]

(同)インデックスアドレス指定

イントラネット[intranet]

インターネット技術を利用した企業内や学校内の情報システム。WWWサーバを社内ネットワークの中心において業務上の情報を交換する。外部からのアクセスは一般向けWWWサーバまでにとどめ、構内用サーバには及ばないようにしている。企業で独自のプロトコルやソフト

を開発しなくても市場に供給されている経済的で安定した製品が利用でき、統一された仕様で外部との連携も管理も容易である。LANに比べ柔軟性があり、管理も容易で、1995年ころから日米の企業で導入、構築するようになっている。インターネット事業を展開するベンダも、イントラネットを支援する技術を積極的に開発中である。JavaやActiveXなどもイントラネットを意識した実行環境の例といえる。

インハウスデータベース[in-house database]

第三者に提供・販売する目的ではなく、企業などが組織の内部で利用するために作ったデータベース、あるいは、商用データベースから社内利用のためにLANに取り込んだデータベース。自社の資料センターが所蔵する業務関連の新聞や雑誌、図書などの資料をデータベース化し、利用要求に即応できるようにしたり、商用データベースの頻用の情報を取り込み、専用端末を用いないでも社内LANやイントラネットですぐに使用できるようにしたシステムが多い。

引用符[quotation／literals quotation／quotation mark]

文字定数を設定するために文字列の前後につける区切り文字。シングル(')とダブル(")がある。数値と数字を区別するシステムでは、数字に引用符をつける場合もある。くくる記号が引用符で、くくられた文字列が文字定数になる。

例えば、COBOLやBASICでは"COBOL"のようにダブルコーテーションマークで、Fortranでは'Fortran'のようにシングルコーテーションマークで、文字列の両端をくくる。またCでは双方を使い分ける。

(参)CSV

インラインイメージ[inline image]

HTMLで書いたページに直ちに表示するイメージデータのこと。比較的データが小さく、多くの人に見せたいイメージはインラインイメージにする。逆に、タイトルを表示しておき、利用者が見たいとクリックしたときに表示するイメージデータを外部イメージという。データが大きく、必要な人にだけ見せたいものは外部イメージにする。

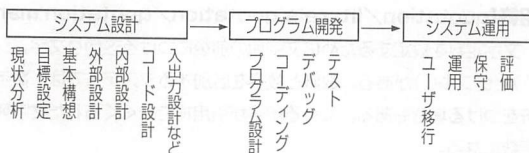
ウ

ウィルス[virus]　(同)コンピュータウィルス

ウォークスルー[walkthrough]

各モジュールのプログラマが同僚の助言を仰ぐことにより、自分が作成したプログラムの仕様や記述の欠陥をさぐること。もとは演劇用語で、役者が監督、演出家に助言を仰ぐ「立ち稽古、リハーサル」の意味である。この場では能力の評価が目的ではないので欠陥を隠す気持ちを起こさせないように管理者を含めないのが通常である。

ウォータフォールモデル[waterfall model]

ソフトウェア開発の諸作業を、基本的に上位工程(上流工程)から下位工程(下流工程)への流れと見なし、各段階では上位工程の成果により作業を進め、下位工程に渡すと考えるモデル。システム規模や特性によって異なるが、一般的には、下表のように区分し、上位から作業を進める。技術的な課題解決は各段階の責任で負う。従来からの主流の開発技法である。問題点として、上位工程で方針を誤ったりミスを犯したりすると、下位工程の作業が難航したり、矛盾が生じたりする。このような事態では上位にフィードバックして上位工程の設計からやり直す。また最終的な段階にならないと、ユーザが直接に目にするシステムのインタフェース部分の姿が見えないという問題がある。(参)プロトタイピング

受入れテスト[acceptance test]

ハードウェア、ソフトウェアの受入れにともなう最終の検査。納入側からは、ユーザの承認を得るためのテストで引き渡しテストという。要求事項を満たしているかどうかをユーザ側のマシンやテストデータで確認する検査で、設置後、ユーザ側の構内で納入側も参加して行う。フィールドテストともいう。

打切り誤差[truncation error]

与えられた数式に実際の数値を代入して計算した値とその数式の真の値との誤差。例えば、(1／3)*3 の真の値は1であるが、有効桁が8桁の浮動小数点数で計算すると、(1／3)=0.33333333　となり、これを3倍しても、0.99999999が得られ、真の値1.0は得られない。何桁まで正しく計算できるかはコンピュータのもっている計算領域の桁数によって決

まる。この桁数より計算精度を必要とする場合は、プログラム上でのくふうが必要である。(関)誤差

埋込み型言語[embedded language]

親言語による適用業務プログラム内の手続きをデータベース操作言語(DML)で記述し、データベース操作を実行できる言語処理系。従来のレコード処理に似た感覚でデータベースのレコード処理ができるようにしたもの。親言語のデータベース処理機能の拡張ともいえる。埋込み型SQLは、既存のコンパイラ言語の中にSQL-DMLを埋め込んでSQLの機能を利用したい場合に使用するタイプのSQLである。

(参)埋込み型言語、SQL

運用[operation]

システムのユーザによるシステムの稼働、システム資源の管理。運用中は、次のような事項を考える必要がある。

(1)性能管理(規定の処理能力の維持・性能の測定や評価)

(2)障害管理(障害の監視・検知・原因や影響の把握・回復や保守)

(3)運用管理(運用の合理化や自動化・運用経費の算出や課金)

(4)機密管理(システムの情報資源の悪用や盗用の防止)

運用テスト[operations test/field test]

開発したシステムが、要求定義の仕様を満たしているか、実際のユーザの運用体制のもとで検査すること。開発用コンピュータと本稼働用コンピュータが異なる場合などには必須の検査である。システムをユーザの環境に移行した後、移行による障害がないかをテストする。通常、従来のシステム作業と並行して運用し、互いの結果を照合しながら行う。

エ

永久ファイル[permanent file]

比較的長期間にわたって保存し、しばしば参照するファイルで、更新しながら継続的に使用する。マスタファイル、基本ファイル、保存ファイルとも呼ぶ。(対)一時ファイル

永久ループ[permanent loop] (参)ループ

エージェント[agent]

ユーザからの直接的な指示によらず、環境や他のプログラムと連携をとり自動的にやるべき処理を判断して行うプログラム。ユーザに代わり、実行に必要なソフトやファイルの情報を収集し他のシステムとの情報交換を行う。

エクステント[extent]

ファイル(データセット)に割当てられたDASDボリューム上の連続した領域、または未割当ての連続した領域。すなわち、磁気ディスクにファイルを記憶させる前に、そのボリュームに確保した必要な大きさの領域であり、確保したエクステントはVTOCに登録しておく必要がある。登録はオペレーティングシステムの仕様にしたがって行う。エクステントが存在する物理的な位置を示す情報をエクステント情報という。

エコー[echo]

キーボードから入力された文字を入力者に確認させるために、画面に表示すること。

エスケープコード[escape code/escape character:ESC]

伝送制御文字のひとつ。ある装置間で情報を転送する場合、このコード(キャラクタ)の次に続くコードの意味を変更する。コードの標準キャラクタセットを拡張するときに用いる。主に2進コードのデータを転送する場合、転送データを制御用文字に拡張する。例えば、「エスケープコードのあとに続く2バイトずつのデータを漢字コードと見なす」のように使う。

エスケープシーケンス[escape sequence]

エスケープコードで始まる文字列。一連のエスケープシーケンスは、通信制御、画面出力制御、プリンタ出力制御、グラフィック制御など、それぞれ独自の体系で特殊な制御に使用する。例えば、画面上ではカーソルをコントロールする、文字の色を変える、文字を反転させることなどができる。また、プリンタでは文字の大きさを変える、アンダーラインを引くなど多くの機能がある。

枝[branch]

木構造で、節(node)と節を結ぶ線をいう。枝は、節と節のつながりを示すほか、問題の図式表現としては方向をもって、因果関係や時間的順序を表したり、大きさ(重み/weight)をもって、節間を結ぶ(または移動する)時間や費用を表すこともできる。(関)木構造

エディタ[editor]

①広義には文章、画像、音楽などのデータ作成のために入力、挿入、消去、移動、複写などの編集処理を行うソフトウェア。編集プログラムともいう。

②狭義には、ソースプログラムや文字データなどの文書を編集するエディタをさす。次のような呼び方をする。

(1)テキストエディタ(text editor)

プログラムやテキスト形式の文章を編集するプログラム。ワードプ

ロセッサに似た文字列の挿入、複写、移動など機能をもつ。ディスプレイ画面全体を使用して編集するスクリーンエディタが主流である。

(2)スクリーンエディタ(screen editor)

カーソル移動キーやマウス、キーボードなどを使用して、画面上に表示したプログラムや文書を編集し、レコードやファイルを作成するプログラム。編集機能はシンプルで、ワープロソフトと異なり文字の修飾的機能はないものが多い。日本語処理はエディタに入力用フロントエンドプロセッサを組み込んで利用する。

エミュレータ[emulator]

あるシステムが、別のシステムと同じデータを受け入れ、同じシステムのプログラムを実行し同じ結果が得られるように、他のシステムを模倣する装置またはプログラムのこと。例えば、パソコンやワークステーションなどのインテリジェントターミナルは、複数のエミュレータを利用することで種類の異なるホストコンピュータの端末として利用することができる。(参)シミュレータ

エラーメッセージ[error message]

システムに与えたコマンド、プログラム、データなどに誤りがあって実行不能の場合、システムが返すメッセージ。誤りの可能性がある場合に出されるメッセージは警告(warning)メッセージという。

(1)入出力や操作に関するエラーは、プログラム実行中にオペレーティングシステムが知らせる。WindowsなどのGUI環境では、異常が発生するとダイアログボックスが開き、状況を報告し、応答を要求する。(類)アラーム、ワーニングメッセージ

(2)ソースプログラムを機械語のオブジェクトプログラムに翻訳または通訳する過程で、文法上の誤りがある場合に、その誤りの種類や原因などを表現する短い文章のこと。(類)シンタックスエラー

エレクトロニックコマース[electronic commerce：EC]

通信ネットワークを利用した商取引。商品やサービスの紹介、受注から決済までの一連の手続き全体を含む形態をさす。

(参)電子データ交換、電子決済、電子マネー

演算子[operator]

表計算ソフトで数式を定義するときや高水準言語でプログラムを書くときに、演算や条件を記述する式の中に使うことのできる演算の動作を表す記号。オペレータともいう。演算子はひとつ(単項)または 二つ(二項)の作用対象(オペランド)をもつ。システムや言語によって形式は異なるが、算術演算子、文字演算子、比較演算子(関係演算子)、論理演算子、代入演算子、条件演算子などがある。

演算装置[arithmetic unit]

中央処理装置の一部分で、演算レジスタをもち、算術演算(四則演算)、比較演算、論理演算などを行う算術論理演算装置の略称。複数のレジスタと演算機構からなる。演算機構は、AND回路、OR回路、NOT回路などの多数の回路を組み合わせたLSIで構成する。コンピュータの演算にはさまざまな方式があり、演算命令の種類もデータの型も多様であるため、算術論理演算装置には、それぞれの型に応じた演算機構やレジスタを使う。算術演算でも固定小数点演算の他、浮動小数点演算、可変長演算、10進演算とそれぞれの演算装置をもつものもある。

演算の順序[sequence of operation] (関)算術演算子

エンティティ[entity]

①OSI参照モデルでは各層にひとつまたは複数存在する機能モジュール。
②データベースの分野では、データの実体のこと。リレーショナルデータベースを理論的に扱うために必要な概念。

エンドユーザ[end user]

直接的、間接的を問わず、最終的な位置でコンピュータシステムを利用する利用者、利用部門のこと。

エンドユーザ言語[end user language : EUL]

エンドユーザが容易に扱えるように作られたプログラム言語や対話型のソフトウェア。メニュー、アイコンなどで操作でき、データ処理をマクロ命令などにより指定できることが特徴である。(参)表計算ソフト

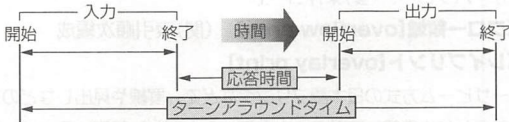
エンドユーザコンピューティング[end user computing : EUC]

情報システムの現実の利用者自身が、自分たちの必要とするシステムを自発的、積極的に構築して利用すること。パソコン上の表計算ソフト、関係データベースソフトなどを利用することが多い。EUCが注目される背景には、分散処理システム、ダウンサイジング、使いやすいパッケージソフトウェアの発達などによりエンドユーザでも開発可能な環境が整ってきたことがある。

エンドユーザデベロップメント[end user development : EUD]

エンドユーザ自身によるシステム開発や保守作業。通常は、アプリケーションプログラムのマクロ機能やエンドユーザ言語を駆使して、個人や所属部門内の小規模システムを開発したり、既成システムを改造したりすることをさす。アプリケーションパッケージをカスタマイズ(帳票形式など機能の一部をユーザ部門の要求仕様に合わせるなど)することを含む。

応答時間[response time]

端末からシステムに要求を送信し、その要求に対する応答データが端末に戻ってくるまでの時間。ネットワークを利用するシステムでは、コンピュータに直結するシステムに比べ、応答時間に影響する要素が多く複雑となる。そのようなシステムを開発する場合、回線保留時間、伝送速度、回線待ち時間など、考えられる要素を洗い出し応答時間を予測することが重要な設計課題となる。(参)ターンアラウンドタイム

応用ソフトウェア[application software／application program]

コンピュータシステムのユーザのデータを直接的に処理するプログラム。応用プログラム、あるいは適用業務プログラムともいう。OSなどの基本ソフトウェア、日本語入力FEPなどのミドルウェアに対比する語。その機能から大きく(1)共通応用ソフトウェア、(2)個別応用ソフトウェアに分けられる。

オーダ[order of complexity]

アルゴリズムの評価に用いる計算量を表す方法のひとつで、問題の大きさ(例えば処理するデータ件数、解く方程式の次数などのパラメタ)によって計算量がどう増減するかを示す式。通常はO(オー)記法によって、問題の大きさをnで表したときの増加の度合で表す。

O(1) ………… nが増加しても計算量は不変
O(log n) …… nが10倍に増加すると、計算量が2倍になる
O(n) ………… nが10倍に増加すると、計算量も10倍になる
O(n log n) … nが10倍に増加すると、計算量が約15倍になる
$O(n^2)$ ……… nが10倍に増加すると、計算量が約100倍になる
$O(2^n)$ ……… nが10倍に増加すると、計算量が約1000倍になる

オーバフロー[overflow／size error]

①演算結果が正しく扱い得る量の範囲を超える状態。通常は算術演算において計算結果の値が大きすぎて規定の記憶領域に入らない状態をいう。文字列演算では、はみ出した語長の部分をさすこともある。あふれ、桁あふれともいう。この状態であることを示すためにオーバフロー用の条件コードを設定し、内部割込みが発生する。

(対)アンダフロー、(参)倍精度

②直接編成ファイルで、ハッシュ法によりあるレコード(A)のキーからアドレスを算出したとき、そのアドレスがすでに別のレコード(B)に占有されている場合、格納場所のないレコード(A)をオーバフローという。

③索引順次編成において、レコード追加により基本データ領域からレコードがはみ出すこと。

オーバフロー検査[overflow check]

演算やデータ転送の処理を行った結果が、レジスタや記憶領域の扱える値の範囲を超えているかどうかを調べるためのハードウェアによる検査。

(関)オーバフロー、(参)条件コード

オーバフロー領域[overflow area]　(関)索引順次編成

オーバレイプリント[overlay print]

レーザビーム方式の日本語プリンタなどで、罫線や見出しなどのフォームをあらかじめ登録しておき、出力データとともに印刷する方式。これにより、あらかじめ用紙に様式を印刷しておくプレプリントが不要になる。用紙の交換頻度が少なくなるほか、用紙在庫の管理も軽減できる。また複雑な帳票類を毎回作成する手間が省ける。(対)プレプリント

オープンシステム[open systems]

オープンは、メーカが製品の仕様や規格を広く一般に公開すること。仕様を公開したOS、基本機能、操作手順、接続法などの標準仕様を公開しているコンピュータシステム全般をさす。

(1)特定の機種やメーカにこだわらず、仕様さえ合致すれば、異なるメーカの情報機器、通信機器を接続できる環境。反対に、同一メーカ、系列メーカの製品間でしか共通性のないシステムを独自(プロプライエタリ)システム、クローズドシステムと呼ぶ。メーカごとにアーキテクチャが異なると、接続できる機種は一定のメーカ製品に限られ、安価で目的にマッチした製品、既得の製品があっても接続できず、異機種を混在したシステム構築は極めて困難だった。

(2)移植性や相互運用性に優れたシステム。UNIXが業界標準のオペレーティングシステムとして急速に普及する、OSIの標準化が順調に進むなど、メーカを越えたシステム間接続の土壌が育っている。特に、通信プロトコル、API、ヒューマンインタフェースの公開や共通化が進んでいるネットワークは、オープンネットワークとも呼ばれる。

(3)異機種コンピュータ間通信を実現するOSI(開放型システム間相互接続)の環境で定義している開放型システム。情報処理と情報転送の両方または一方を実行でき、コンピュータ、関連ソフトウェア、周辺

装置、端末、操作員、情報転送手段などから構成される独立したシステム。OSI参照モデルでは開放型システムの中を7階層に階層化している。

オフィスオートメーションシステム[office automation system：OA system]

オフィス内の情報処理機器を機能的に結びつけ、オフィス業務の合理化、省力化を図り、生産性の向上や経営の改善を目指すシステム。パソコンを端末とし、プリンタ、電子ファイル、ファクシミリなど、各種OA機器をLANで結び、情報を総合的に管理する統合OAシステムが増えている。

オブジェクト指向[object oriented]

処理の手順(制御構造)とデータ構造を中心とした手続き型手法とは異なり、処理対象(オブジェクト)そのものを中心に展開する情報処理の考え方。GUI環境は、オブジェクト指向の考え方から生まれた。データとそれに関する操作や手続き(メソッド)をひとまとめにしてオブジェクトという形式の単位にまとめ、多数のオブジェクト間でメッセージをやりとりしながら全体の処理を進める。処理はオブジェクトへの要求という形で表現する。オブジェクトは他のオブジェクトに関知しないため、オブジェクトの独立性が高まり、プログラムが単純化できる。

オブジェクト指向言語[object oriented language]

処理するデータを重視し、プログラム中で積極的にデータを中心に記述できるタイプのプログラム言語。このような言語の例としてSmalltalkやC++、Object Cなどがある。従来の手続き型言語によるプログラミングでは、もとになるデータを、手順を追って処理し、目的の結果(出力)を得るため、プログラミングの中核はデータを処理する手順を考えることにあった。またハードウェアの実行順序もプログラムの記述順に密着している。また、データはファイルという概念でプログラムとは別物と意識することが多い。

これに対し、データと手続きに相当するオブジェクトは物、目的、対象であり、処理はオブジェクトにメッセージを送り、オブジェクトは返事(値)を返すという形で進行する。返事は他のオブジェクトへのメッセージになり得る。

オブジェクト指向プログラミング[object oriented programming]

モジュールの階層化と独立化、部品化を積極的に意識して、オブジェクト指向言語を利用して行うプログラミングの思想、またはその工程、作業。手続き型のプログラム言語によるプログラミングが、データ処理の手順を考えることを中心にしたのに対比し、処理するデータを重視し、プログラム中で積極的にデータを中心に記述できるタイプのプログラミング手

法である。イベント駆動のウィンドウシステムは、オブジェクト指向プログラミングによくあう応用分野である。オブジェクト指向の考え方はクライアントサーバシステム、分散オペレーティングシステムにも適用されている。(参)オブジェクト指向

オブジェクトプログラム[object program]

ソースプログラムをアセンブラやコンパイラで翻訳し、生成した機械語コードのプログラム。目的プログラム、ターゲットプログラムともいう。オブジェクトプログラムは、リンケージエディタを介して実行可能なプログラムにする。(参)コンパイル

オフライン[offline]

周辺装置などがCPUと論理的に直接には結ばれていない状態。通信回線などを利用して得たデータやプログラムをいったん磁気ディスクなどの記憶媒体に記録してから、あらためてコンピュータに入力したり出力したりするシステム。オンラインシステムの普及以前の一般的なシステム。オンラインに対比してオフラインという。(対)オンライン

オペランド[operand]

①コンピュータの命令語の構成要素で、命令の対象となるデータを指定する部分をいう。通常は処理するデータの位置を指定するレジスタの番号や主記憶装置の番地を指定するが、データそのものの場合もある。例えば、A番地のメモリに記憶せよという命令では、アドレスAがオペランドとなる。(関)命令、(参)アドレス部

②広くは演算(operation)の対象となるデータ(被演算数ともいう)。単項演算子はひとつ、二項演算子はふたつ、三項演算子は3つのオペランドをもつ。x+yという二項演算では、+が演算子(オペレータ)で、xとyがオペランドである。オペランドは変数名だけでなく、表計算ソフトではセル番地などで指定することもできる。(参)演算子、項②

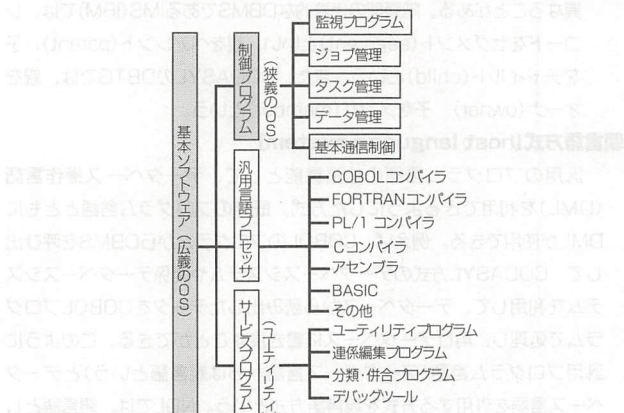
オペレーション[operation]

①コンピュータの1命令で規定する1動作または1演算。計算機による1オペレーションの実行を1ステップ(step)ともいう。

②コンピュータの操作。計算機に処理させるひとまとまりの処理過程をジョブと呼び、ジョブのスケジュールを行い、計算機を操作しながら、ジョブを実行していくこと。これらの大部分が自動化されてため、狭義には、記憶媒体のセット、プリンタへの用紙のセット、それらの機器を起動するためのコマンドの投入などのマニュアル操作をさす。

オペレーティングシステム[operating system：OS]

コンピュータ資源、ネットワーク資源を管理し、効率よく働かせるため

のソフトウェアの集まり。ユーザの処理プログラムの実行中にも必要な種々のサービスや状態監視を常時行う。狭義には制御プログラムをさす。

【オペレーティングシステムの機能とねらい】

コンピュータが処理プログラムを実行するためには、その前後のための作業が必要となる。また、処理プログラムの実行中にも種々の処理が必要であるが、これらのサービスや監視を行う。システムの処理能力(スループット)の向上、応答時間の短縮、信頼性・可用性の向上、ヒューマンインタフェースの向上などが、オペレーティングシステムのねらいである。このため基本ソフトウェアということがある。

【代表的なオペレーティングシステム】

従来から大型汎用コンピュータについては、各コンピュータメーカが独自のOSを開発していたため、OSの互換性はとぼしかった。しかし、ワークステーションやパソコンについては、どのメーカの機種でも使用できる標準OSが出現するようになった。その代表的なものに、MS-DOS、Windows、UNIX、OS／2などがある。また、分散処理システムの普及とともに、NetWare、Windows NTなど通信ネットワークを強く意識したネットワークOS(NOS)も現われている。

重み[weight]

①位取り記数法の各桁に掛ける係数。(参)基数記数法

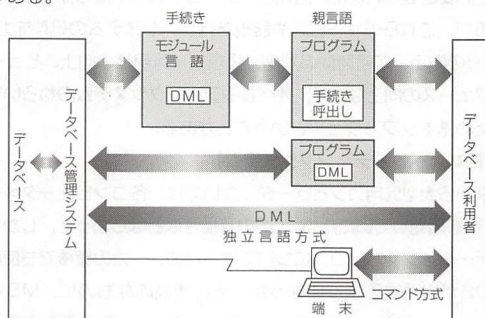
②複数の要素をもつシステムを評価するとき、各要素の影響力や効果を考えてその要素の値に掛ける係数。重み係数ともいう。

親[parent／owner]

①木構造における上位の節。下位に対しては複数のポインタがあり得る。下位の節を子(child)と呼ぶ。

②親子集合型(set type)のデータベースで、相対的に上位のレコードを親レコード、下位のものを子レコードという。この呼び方はシステムで異なることがある。階層型の代表的なDBMSであるIMS(IBM)では、レコードをセグメント(segment)といい、親をペアレント(parent)、子をチャイルド(child)という。また、CODASYLのDBTGでは、親をオーナ(owner)、子をメンバ(member)という。

親言語方式[host language system]

汎用のプログラム言語の拡張機能として、データベース操作言語(DML)を利用できるようにした方式。既存のプログラム言語とともにDMLが使用できる。例えば、COBOLのプログラムからDBMSを呼び出して、CODASYL方式のデータベースシステムや関係データベースシステムを利用して、データベースから読み出したデータをCOBOLプログラムで処理し、再びデータベースに書き戻すことができる。このように汎用プログラム言語(データベース言語からは親言語という)とデータベース言語を併用する方式を親言語方式という。NDLでは、親言語として、COBOL、Fortran、PL/I、Pascalを指定している。親言語方式には、(1)モジュール言語を利用する方式、(2)埋め込み言語を利用する方式がある。

親子集合型[set type]

データやデータセットのネットワーク構造で、一方が他方の従属関係にあるとき、その組を親子集合型という。親子集合型のレコードの集まりを親子集合という。相対的に上位のレコードを親レコード、下位のものを子レコード、最上位のものを根レコード(root record)という。

音声データ[audio data]

人の声、音楽、自然の音などをディジタル化したデータ。ディジタル化により合成・加工・蓄積・圧縮・伝送などがコンピュータにより処理できるようなる。Windows 95/98ではWAVファイルが標準となってい

る。アナログデータの処理過程で発生しやすい雑音の混入や音品質の劣化に対して非常に強い性質をもつ。ディジタル化技術の発達によって、より自然に近い音声を再現できるようになった。音声データの圧縮はADPCM方式で行うのが一般的である。

オンデマンドサービス[on demand service]

端末からユーザが要求した情報をシステムが即座に提供するサービス。マルチメディアを前提に、ビデオをネットワークを通じて送り出すビデオオンデマンド、必要なニュースだけをネットワークを通じて提供するニュースオンデマンド、オンラインショッピングなどがオンデマンドサービスに含まれる。システムがあらかじめ決められた情報を繰り返して提供するのではなく、ユーザのその場その場のさまざまな求めに柔軟かつ容易に応じることが、このサービスに欠かせない考え方である。

オンデマンド処理[on demand processing]

要求時点で即時に行う処理。(同)リアルタイム処理

オンライン[online]

周辺装置や端末がコンピュータに直接電気的に結びついて、データのやりとりができる状態(無線通信も含む)。これに対し、データを一方から他方へ送る途中で人間の操作が必要な状態をオフラインという。

(対)オフライン

オンラインシステム[online system]

ホストコンピュータと遠隔地にある端末装置(リモート端末)とが通信回線で結ばれ、両者間でデータの交換を行うシステム。

オンライン処理[online data processing]

オンラインシステムで、データの収集や処理、交換を行うこと。オンラインシステムの処理形態として、大きくふたつに分けられる。

(1)データを一定時に処理するオンラインバッチ処理

(2)データが発生した時点で即座に処理するオンラインリアルタイム処理

またその利用形態には、データ収集、メッセージ交換、照会、応答、遠隔ジョブ入力、タイムシェアリングなどがある。

オンライントランザクション処理[online transaction processing：OLTP]

端末から頻繁に発生する処理要求(トランザクション)に対し、ある一定時間内に処理(データベースアクセスを含む)を完了し、応答する処理をオンラインで行うこと。OLTPは非常に信頼性と性能が高いシステムを必要とするため、フォールトトレラント型のコンピュータをFEPに使用して処理するシステムが増えている。トランザクションが通信設備や通信処理に影響を与える要素には、まず入出力メッセージの種類、長さ、量、分布な

どがあげられる。ひとつのトランザクションは処理の途中のまま放置しておくことができない性質のものである。そのため、トランザクション処理では、通信障害に対する万全の対策が必要となる。

オンラインバッチ処理[online batch processing]

基本的に一定時に行うバッチ処理であるが、データの収集は通信回線を利用し、処理結果も通信で行う処理形態。

オンラインプログラム[online program]

オンラインで結ばれたコンピュータと端末装置との間で、データのやりとりを行い、ひとつの処理が比較的短時間で終わるように作成した応用プログラム。いっぽう、さまざまな処理をひとまとめにして処理するように作った応用プログラムをバッチプログラムという。

オンラインマニュアル[built-in documents]

電子化マニュアル。操作説明書に相当する情報をシステムの内部に組み込み、利用者が必要になった場合、ヘルプ機能として画面上に提供したり、ガイダンス情報として参照できるようにしたマニュアル。目次検索、キーワード検索などもできる。印刷物として膨大なマニュアルを作成する経費や資源を節約すると同時に、ユーザが必要なつど必要な部分だけを参照できれば十分という考え方に基づいている。なお、初期設定、基本機能、基本操作については印刷物のマニュアルが添付される場合が多い。

オンラインリアルタイム処理システム[online real time processing system]

通信回線を利用して、遠隔地にある複数の端末(ユーザ)からホストコンピュータにアクセスでき、ホストコンピュータは、端末の要求を即時に(厳密には短時間に)処理して端末側に返送するシステム。対話型システムが多い。オンラインリアルタイム処理では、端末から対話式で応答を行う方式が多く、代表例には、列車や飛行機などの座席予約システム、銀行のCDやATM制御システム、送電線や変電設備の障害を監視する電力系統制御システム、交通量から信号機や道路表示板の制御をする交通管制システム、データベース検索システムなどがある。

カーソル[cursor]

①適用業務プログラムからデータベースのデータをアクセスする場合のデータベース操作言語の概念。1個のキーの値(あるいはそのキーをもつタプルの番地)を蓄えることができる一時記憶装置。プログラムとデータベースの間でデータ引き渡しをする抽出条件付きの行である。

カーソルを含むプログラムを実行すると、リレーショナルDBMSにより一時的に結果の表が作られ、この表が順編成ファイルに相当するので、適用業務プログラム側からは、順編成ファイルをオープンしたとみなすことができる。

②システムが次の文字入力を受けつけるときにディスプレイ画面にその位置を示す可視の記号。文字を入力すると、カーソルの位置にその文字を表示する。

③ディジタイザやタブレットなどの座標入力装置の板面上で座標指定のために用いる補助具。

カード型データベース[card type database]

1レコードのデータを1枚のカード、またはシートに見立てて記録したファイルを扱うデータベース。単一構造のファイルしか扱わないため、小規模で単純な構造のデータ管理に向いている。いくつかの項目での検索、項目間の計算、条件抽出、集計、分類、並べ換えなどの機能をもつ。リレーショナルデータベースソフトの中には、1レコードを1画面にカード形式で表示する機能をもつものもある。(参)リレーショナルデータベース

ガーベイジコレクション[garbage collection]

ガーベイジ(使用済みのデータ)を除去し、現在使用している領域を再編成して大きな空き領域を確保し、記憶装置の有効利用を図ること。特にLISPやProlog系の言語を使用する場合は不可欠である。(参)再編成

外字[external character]

システムやフォントセットが標準装備していないため、ユーザが独自に作成、登録した文字や記号のパターン。パソコンやワープロでは、JISの第2水準までの漢字や記号に加えメーカが独自に選定した拡張漢字をシステムに登録している。標準装備以外の漢字や記号、文字を使用する場合、ユーザが外字エディタの画面上で文字のでき具合を確認しながらパターンを独自に作成し登録するか、フォントメーカが作成した文字セットを購入する必要がある。この場合、外字は区点番号や登録した読みで呼び出して使用する。なお、こうした外字は他のシステムでは使用できないことがある。

カ

解釈プログラム[interpreter]　(同)インタプリタ

解析木[parse tree]

コンパイラの構文解析過程で、構文要素の解析により作られる木構造の表現。構文解析後の処理に適した構造といえる。構文木とほぼ同義。

階層型データベース[hierarchical database]

データベース内のレコード間の関係が階層型(上位と下位のレコード間の関係を1：Nに限る。木構造モデルともいう)であるデータベース。

階層構造[hierarchy structure]

広義には、ハードウェア・ソフトウェア・データなどの資源の特性に注目して複数の水準に分け、木構造状に配し、システム全体の効率を高めるように構成したときの構造。各階層はおのおのの役割とする管理・制御の機能をもつ。以下のような例がある。

(1) 異なる特性をもつ記憶装置を階層的に配して利用する。
　(参)キャッシュメモリ、バッファ記憶装置

(2) システムの処理機能を分割し、階層的に設計する場合を特に構造化設計という。プログラムの場合、モジュール単位に分割し、階層的に配して構成する場合を特に構造化プログラミングという。
　(参)モジュール

(3) データベースやファイルなどで、データやレコード間の特性に注目して基本項目(これ以上には細分化できないデータ)をまとめて集団項目とする際に階層的に構成する場合、データは階層構造をもつという。
　(参)データベース、木構造、ファイル管理

解凍[data expansion／data decompression]

圧縮したファイルをもとへ戻す処理。(参)データ圧縮

解読[decode]

命令レジスタに取り込んだ命令を、命令部とアドレス部に分けた後、命令の内容を実際の動作に直したり、実効アドレスを算出すること。
(参)実行サイクル

概念スキーマ[conceptual schema]　(参)スキーマ

外部記憶装置[external storage]

プログラムやデータを大量に保存しておくために主記憶装置の外部に置く記憶装置。主記憶装置を内部記憶装置と呼ぶことに対応する。外部記憶装置にアクセスするには、入出力インタフェースを通して行う。

外部記憶装置には、汎用コンピュータ用の磁気テープ記憶装置、磁気ディスク記憶装置、さらにパソコン用のフロッピーディスク装置、ハードディスク装置、CD-ROM装置、光磁気ディスク装置(MO)などがある。これらは周辺装置として本体の外部に置く外付けタイプと本体の拡張スロッ

トなどに装着する内蔵タイプがある。内蔵されていても原理的には外部記憶装置である。

外部コマンド[external command] (参)コマンド

外部スキーマ[external schema] (参)スキーマ

外部設計[outline design]

システム設計の領域は、外部設計(概要設計)と内部設計(詳細設計)とに大別できる。外部設計は、システムのおおまかな機能と構成を設計すること。予備設計あるいは基本設計ともいう。現状分析の結果に基づき、その問題点を明らかにしたうえで、要求仕様をまとめ、実現可能なシステムの基本構想をこの段階で作成する。新システムで、どのようなデータを、どう集め(入出力情報の設定)、いかに処理するか(基本機能の概要)、外部システムとのインタフェース、安全性や信頼性のレベルなどを決める。

外部ソート[external sort]

磁気ディスク装置などの外部記憶装置を用いて行うソート。大量のデータをソートする場合、量に見合う大きな作業エリアが必要であるが、主記憶装置だけでは容量不足となる場合、磁気ディスク装置などの外部記憶装置を用いて分類を行う。COBOLには大規模なファイルを整列するために、SORT文が用意されている。他の言語では、外部ソートを行う場合は、メーカ提供のユーティリティプログラムを使うことが多い。

(対)内部ソート

外部表現[external representation of data]

ソースプログラム上の制御文やデータの表現のことで、テキスト形式である。通常1バイトの文字が使用される。英数字以外の地域化の指定がある場合は、多バイト文字や拡張文字集合の文字が使用される場合もある。

(参)文字集合、(対)内部表現

外部ラベル[external label/sticker] (関)ラベル

外部割込み[external interruption : EI] (関)割込み

概要設計[outline design] (同)外部設計

カウントキーデータ形式[count key data architecture]

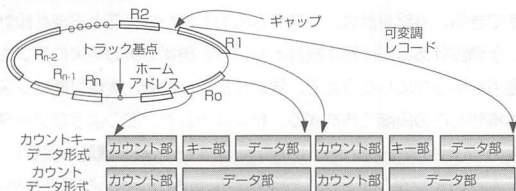
磁気ディスクに索引順次編成ファイルを記録するときの形式。すなわち、カウント部、キー部の後にデータレコード部が置かれる。ギャップがカウントデータ形式にくらべ多くなるため、トラック当たりの記憶容量は小さくなる。

カウントデータ形式[count data architecture]

磁気ディスクに順次編成、直接編成、VSAM編成のファイルを記録するときの一般的な形式。すなわち、カウント部の直後にデータレコード部が置かれる。

カ

カウント部[count area]

磁気ディスク上のレコードが存在するアドレスを示す部分。磁気ディスク上のトラックに記録されるレコード(データレコード部)の前に記録される。アドレス部(address part)ともいう。アドレスを示すシリンダ番号(CC)・ヘッド番号(HH)・レコード番号(R)、キー長、データ長、誤り訂正符号で構成される。

拡張子[extension]

ファイル名の一部であり、ファイル名の後ろに、ファイルの種類や属性などを表したもの。ファイル名は、MS-DOSの場合は半角 8文字以内、Windows95の場合は半角256文字以内、ピリオドの後の拡張子は半角 3文字以内と定めている。任意に付けることができるが、システムが解釈する定型的な拡張子もある。MS-DOSやWindowsの例としては「. COM」、「. EXE」(実行可能なプログラムの意味)、「. SYS」(デバイスドライバなどのシステム側で使用)、「. BAT」(記述した命令を順番に実行するバッチファイル)などがある。

拡張2進化10進コード[extended binary coded decimal interchange code] (同)EBCDIC

拡張文字集合[extended character set]

基本文字集合に多バイト文字や地域化文字(文化圏固有文字)を加えたもの。(関)文字集合

加工セグメント[manipulating segment]

(参)処理・加工セグメント

仮数[mantissa]

浮動小数点表示において、表現される実数を決めるために、基数のべき乗にかける数。例えば、0.1234×165では0.1234が仮数。浮動小数点レジスタの仮数部には1234が、指数部には 5が入る。

仮数部[mantissa field]

浮動小数点レジスタに浮動小数点数を記憶する際の、仮数を記憶する桁部分。通常、1語~2語の中を符号部、指数部、仮数部に分けて記憶する。通常、仮数部の最上位桁が1になるように正規化されるが、これを暗黙の了解とし、最上位桁の1を省略する場合もある。(関)指数

カ

カスタマイズ[customize]

多数のユーザ向けに設定されているハードウェアやソフトウェアの機能を、特定のユーザの運用形態や使用目的に合わせて変更すること。

(1) ハードウェアでは、既製のICの回路を変更して特殊なICを製造することや企業の大量注文に応じてメモリや電源などを増設したパソコンを製造することなど。

(2) ソフトウェアでは、入力時のキーの役割や出力画面のデザインを可能な候補から選択すること、給与計算プログラムや在庫管理プログラムなどのアプリケーションを自社の制度やシステムに合わせて仕様変更することまでを含む。(関)パッケージソフト

仮想記憶[virtual storage：VS]

物理的な主記憶の限界を越え、より大きな論理的な主記憶空間を作り出す方式。主記憶装置と補助記憶装置の連携をとって、多量のデータや大きなプログラム、多重プログラミングのために複数のプログラムを処理する技法。プログラムやデータをページという単位に分割し、補助記憶装置にいれておき、当面必要なページだけを主記憶装置にロードし実行する。必要なページが主記憶にない場合は、不要なページを補助記憶に戻し、あらためて必要プログラムをロードし実行する。この方式が可能であるためには、プログラムはあらかじめ主記憶装置のどこにロードされても実行可能なように(リロケータブルに)作っておかなければならない。仮想記憶に対比してハード的な主記憶装置を実記憶装置という。仮想記憶方式により、主記憶装置の記憶容量より大きなプログラムも実行できるようになった。

型[type／data type]

データや関数の性質および内部表現。数値データ型としては整数型、実数型、複素数型などの種類がある。型によって許される演算の種類が異なる。実数はその精度により、単精度実数型、倍精度実数型、多倍精度実数型などがある。

(1) **型宣言**[typing／data type declaration]

Fortranの用語で、変数、定数、配列などの英字名の型を型宣言文で指定すること。型宣言をしないと、英字名の頭文字から処理系が判断した暗黙の型が与えられる。(参)暗黙の型宣言

(2) **型指定子**[type specifier]

Cの用語で、型は変数等のデータオブジェクトの形式。型指定子は、型と大きさを表現する指定子と、構造体・共用体・列挙型の指定子のこと。これらの型指定子の組合せで型を表現する。

型変換[data type conversion／type conversion]

プログラム実行中にデータの型を別の型に明示的に、もしくは暗黙に変

換すること。データ変換ともいう。次の2つの場合がある。

(1)明示的な型変換：組込み関数やキャスト式で行う

(2)暗黙の型変換：代入式や算術式の演算の際に式の右辺の型に合わせて、左辺の型が変わる。Cでは、拡張ともいう。

カタログ[catalog／catalogue]

記憶媒体上のファイルに関する情報(ファイル名やファイル編成、レコード長、記憶した先頭アドレス、参照情報、保護情報など)を一括して管理する記録。区分編成ではディレクトリ、索引順次編成では索引がこれに相当する。ファイル操作を行うプログラムを実行する際に、OSに与えておく情報である。(参)登録簿、ディレクトリ、索引

稼働率[availability／operating ratio]　(参)信頼性評価尺度

かな漢字変換[kana-to-kanji transformation]

ワープロやコンピュータシステムに漢字を入力する際、かな、またはローマ字で入力した読み(かな)を漢字かな混じり文に変換すること。日本語には同音異字が多く、簡単、直接の変換方法がくふうされてきた。

カフェテリア[cafeteria]

本来の意味は自分で料理を運ぶ簡易食堂。情報処理では、クローズドバッチ処理とオープンバッチ処理の中間にあって、ユーザ自身がバッチの登録を行い、処理結果を受け取るが、コンピュータの操作は専任のオペレータが行う方式。オペレータの仕事の一部をユーザに開放した方式といえる。

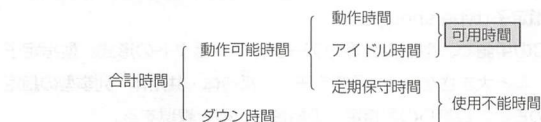
可変長レコード[variable length record]　(関)レコードタイプ

画面設計[screen design／screen layout]

入出力設計の工程で、ユーザがデータを入出力する画面を設計すること。表示するデータ項目の並び、レイアウトを決定する。画面サイズを考慮し、操作しやすく設計する。特に、入力操作をともなう画面は、誤入力を誘うことがないように、理解しやすく、扱いやすくする配慮が必要である。

可用時間[available time]

利用者の立場からみて、コンピュータの機能単位を使用できる時間。コンピュータが実際に使われているかどうかに関係なく、使用が可能な状態にある時間をいう。(参)信頼性評価尺度

- 合計時間
  - 動作可能時間
    - 動作時間 ┐可用時間
    - アイドル時間 ┘
    - 定期保守時間 ┐使用不能時間
  - ダウン時間 ┘

可用性[availability]　(参)信頼性評価尺度

仮引数[parameter／dummy argument]　(関)引数

下流CASE(lower CASE tool) (参)CASEツール

カレント[current]

ある時点でユーザやプログラムの操作の対象となっているものをさす。

(1)カレントディレクトリは、ファイルアクセスのためにユーザが選んだディレクトリ。UNIX系OSやMS-DOSの階層的ファイルシステム(FMS)では、起動時点で一般的にルートディレクトリ(あるいはユーザ用ホームディレクトリ)に位置づけられる。ユーザはパスを移動して、自分に必要なファイルの所在する階層的ファイルシステム中のディレクトリを移動してアクセスできる。

(2)カレントドライブは、ワークステーションやパソコンに接続されている複数台のディスクドライブのうち、ユーザの操作の対象になっているファイルが属するドライブ。ボリューム指定がされていない場合は、カレントドライブがアクセス対象となる。

間隔/間隔文字[space/space character] (同)空白文字

関係演算[relational operation] (同)関係代数

関係演算子[relational operator] (同)比較演算子

関係代数[relation algebra]

リレーショナルデータベースの表に対する操作を定義する演算系の集まり。この演算を適用することによって、必要な情報を取り出すことができる。E.F.Coddは、選択、射影、結合、除算の4つの関係演算と従来の集合論にあった4つの集合演算(直積、合併、共通、差)を提案した。これらを整理して、リレーショナルデータベースで使用するデータ操作言語になった。次の3種類の基本演算がある。

(1)選択(selection):元になる表から必要なデータをある条件を設定して取り出す作業。射影は列単位にデータを取り出すが、選択は行単位でデータを取り出す。制約ともいう。

(2)射影(projection):元になる表の中の特定の項目(属性)だけを指定して、それだけの新しい表を作る機能。列単位でデータを取り出す。

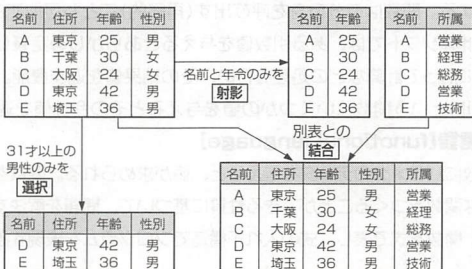
(3)結合(join):元になる表と比較する表とで共通する属性を調べ、設定

名前	住所	年齢	性別
A	東京	25	男
B	千葉	30	女
C	大阪	24	女
D	東京	42	男
E	埼玉	36	男

名前	年齢
A	25
B	30
C	24
D	42
E	36

名前	所属
A	営業
B	経理
C	総務
D	営業
E	技術

名前	住所	年齢	性別	所属
A	東京	25	男	営業
B	千葉	30	女	経理
C	大阪	24	女	総務
D	東京	42	男	営業
E	埼玉	36	男	技術

名前	住所	年齢	性別
D	東京	42	男
E	埼玉	36	男

した条件を満たすものを取り出し新しい表を作成する。射影と選択を一度に行うような機能にあたる。

カ

関係データベース[relational database：RDB]

関連するデータを、行と列に展開した二次元の表（テーブル）形式で取り扱うデータベース。リレーショナルデータベースともいう。人間が日常的に使う表の形式にイメージが近く、わかりやすく使いやすい点が特徴である。表と表との関係づけを、データの値によって行う。データ構造表現が柔軟で、設計や拡張、変更が容易であるうえ、データの独立性が高い。

データを表名、項目名、条件を指定して取り出して加工でき、表の操作は集合論の関係代数を利用して演算で行う。（関）関係代数

漢字コード[kanji code／Chinese character code]

（同）JIS情報交換用漢字符号、（参）EUC②、UNICODE

換字式暗号[substitution cipher] （関）慣用暗号系

監視プログラム[supervisor／supervisory program]

（関）制御プログラム

環状リスト[circular list] （関）リスト構造

関数[function]

①ある変数の値が、他のいくつかの変数の値を与えたとき、一意に決まる関係にあるとき、この関係を関数という。数学では、例えば、x、yの値でzが定義できるとき、$z=f(x, y)$のように表現する。

②Fortranでは、メインプログラムから呼び出して使用できるまとまった機能をもったルーチンをいう。プログラムでは、関数に名前をつけて宣言したり、手続きとして定義することができ、実行文を簡潔に書くことができる。（例）組込み関数、関数副プログラム、サブルーチン

③Cでは主たるプログラムを関数の呼出しとして記述する。コンパイルの最小構成単位である。関数の本体はCの命令をひとつ以上含む文の集合からなる。関数は使用される前に宣言するか定義されていなければならない。関数は呼び出し元から引数を受け取り、関数処理の実行後、返却値を返す。ただし、引数のない関数や返却値のない関数も存在する。関数は自分自身を呼び出す（再帰的）ことも可能である。

④表計算ソフトでは、ある引数値を与えるとあらかじめ定義してある手順に沿って計算などの処理を行い、その結果値を返す機能。例えば、SUMという関数にいくつかの値を与えるとその合計値を返す。

関数型言語[functional language]

関数にパラメタのデータを与えると、値が求められる。関数を合成して新たな関数をつくることができる性質に基づいて、問題を解決する方法がある。関数を式で表し、式の入れ子構造でプログラムを表現する言語を関

数型言語という。プログラミングは関数を組み合わせて目的の関数をつくる。LISP、APLなどが代表的な関数型言語である。

関数定義[function definition]

関数が行う処理を記述すること。処理系があらかじめ用意していない関数でもユーザが定義すれば使用できる。

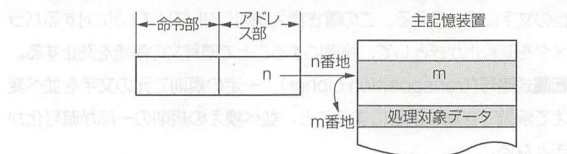
関数副プログラム[function subprogram]

Fortranの用語。FUNCTIONステートメントで始まるFortranプログラムの外部関数。式の中で引用すると、呼び出したプログラムに関数の値を返す。

関数プロトタイプ[function prototype]

Cの用語。関数とその引数の宣言のこと。関数呼出しに先だって関数プロトタイプの宣言を行う。関数プロトタイプ宣言を元に、コンパイル時に実引数と仮引数のチェックが行われる。ただし標準ライブラリ関数に関しては、ヘッダファイル内に関数プロトタイプが宣言されているので、関連するヘッダのincludeを行えばよい。

間接アドレス指定[indirect address／indirect addressing]

アドレス指定の一方式。命令語のアドレス部で、処理対象データを格納する物理アドレスを記憶する場所を指定する方式。物理アドレスを記憶す

る場所は、レジスタや主記憶装置の別の場所である。
(対)直接アドレス指定

間接アドレス方式[indirect addressing]

直接編成ファイルでレコードを指定する一方法。レコードのキー値をそのまま、シリンダ番号(CC)、ヘッド番号(HH)、レコード番号(R)に当てて直接的にアドレスを指定する直接アドレス方式に対比する語。間接アドレス方式は、レコードのキー値にある演算を施し、その結果を利用してレコードのアドレスとする方式。ここで、キー値をアドレスに変えることをアドレス変換(address translation)またはランダマイジング(randomizing)という。

どのような演算を施すかはキー値の分布によるが、除算法・数字分析法・基数変換法、重ね合わせ法などでアドレスが均一に、かつ無駄なく分布するようくふうする。しかし、あるレコードのキー値に演算を施した結果が別のキーのレコードアドレスと同じになることがある。その場合これ

らのレコードをシノニムレコードという。
(対)直接アドレス方式、(関)シノニム

完全性[integrity]

システムの信頼性の概念のひとつ。データは日々発生し、システムで処理し、ファイルとして保存するが、データ、システム、処理プログラム、ファイルのすべてが正確で信頼できるものでなくてはならない。しかし、これらをそこなう危険は常に存在するため、データやファイルの正確さ、完全性を検証しなければならないとする考え方。データのチェック、プログラムのテスト、出力のチェック、障害の回復、破壊したデータの復元等の技法に関連する。(関)RASIS

慣用暗号系[conventional encryption system]

暗号化と復号に同一のかぎを用い、そのかぎを秘密にする暗号方式の総称。共通かぎ暗号方式、または秘密かぎ暗号方式ともいう。かぎは64ビットのものが一般的で、公開かぎ暗号系に比べてかぎの長さが短く、暗号化速度も速いのが特徴。代表的なものに米国のDES方式、NTTのFEAL方式、ヨーロッパのIDEAなどがある。モデル的な慣用暗号系(共通かぎ暗号方式)には次のような方式がある。

(1) **換字式暗号**(substitution cipher):メッセージ内の文字をある規則で他の文字に置き換える。この置き換え規則(アルゴリズム)に対するパラメタを暗号化かぎとして、秘密にすることで暗号文の解読を防止する。

(2) **転置式暗号**(transposition cipher):一定の規則で元の文字を並べ変えて解読できないようにすること。並べ換えの規則の一部が暗号化かぎとなる。

キ

偽[false]

論理変数のとり得る値のひとつ。「真」の対語。ある条件のもとで命題や述語が成立しないとき偽という。(対)真、(参)論理変数

キー[key]

データ項目中に含む1個以上の文字であって、そのデータを認識したり、その用途を制御したりするために使えるもの。または、レコードを検索したり、整列したりする場合の手がかりとなるデータ。データコードやテーブルのインデックス、ソートキーなど。

キー順データセット[keyed sequenced data set:KSDS]

仮想記憶のもとで、索引編成ファイルと同様に順アクセスと索引による

直接アクセスが可能なデータセット。KSDSは索引を作成するインデックスコンポーネント(索引構成要素)と、レコードを記憶するデータコンポーネント(データ構成要素)からなる。

KSDSに対するアクセスは、順アクセス、直接アクセス、動的アクセスが可能である。直接アクセスの場合、アクセスするキー値を指定すると、索引セットから順序セットを特定し、さらに順序セットからレコードのあるCIを特定する。さらにCIを探索して目的のレコードにアクセスする。

KSDSに対する処理では、更新・追加・削除が可能である。レコードの追加の場合は、空きスペースを使って、キー値の示す位置に追加される。例えば右図でキー値が035というレコードを追加する場合は、030のレコードの次に追加され、040のレコードは035のレコードの後に移動する。また、追加・削除により、CI中の最後のレコードのキー値が変わった場合、索引の内容も更新される。(関)VSAM

キー部[key area]

索引順次編成ファイルのレコードを記録する場合、カウント部に続けて設けるキーを記録する部分。ディスクアドレスによらず、キー項目で指定レコードを早く探す。キー部は、キーと誤り訂正符号で構成される。

記憶スキーマ[storage schema]

データベースの構造と内容の記述をスキーマという。スキーマに記述された内容を記憶装置に実現する役割を果たす。ANSIの3層スキーマの内部スキーマに対応する。CODASYLのDBTGでは、データベースの論理構造をスキーマとサブスキーマの2階層でとらえているが、さらにそれらのスキーマに記述された内容を計算機上にどのように実現するかを記憶スキーマで記述する。

記憶密度[packing density]

記憶媒体の性能を示すめやすで、単位長、単位面積または単位体積当たりの記憶媒体に記録可能な情報量。磁気テープの場合は、単位長さ当たりのビット数またはバイト数で表現する。磁気ディスクの場合は、1トラックに記憶するビット数またはバイト数で表現する。
(関)BPI

記憶容量[storage capacity]

一単位の記憶装置、記憶媒体にたくわえられる情報の最大量。記憶装置、記憶媒体の規模の大小を表す指標ともなる。通常バイト単位で表現する。補助単位としてk(キロ:10^3)、M(メガ:10^6)、G(ギガ:10^9)などを使う。

ギガ[giga:G]

10^9(10億)のこと。ただし情報処理では、2^{10}=1024を1000と見な

して、ギガとは1024×1メガをさす場合もある。

機械語[machine language]

コンピュータが直接に実行可能な命令コードの集合、または命令コードで書かれたプログラム。2進コードで表現する。事実上、人間が高水準言語で書いたソースプログラムをアセンブル、またはコンパイルした結果として出力する命令コード(オブジェクトプログラム)である。この形式は、コンピュータの機種ごとに異なり、相互の互換性がない。

機械向き言語[machine oriented language]

機械固有の機械語に1対1に対応する記号命令語を主とするプログラム言語。アセンブラ言語が代表的。Fortran、COBOLなどの人間の言語に近い高水準言語に対比される。(類)アセンブラ

機械割込み[machine check interruption] (関)割込み

基幹システム[backbone system／corporate system／enterprise system]

全社的な情報システムの中核をなす主要なコンピュータシステム。部門別、個人別の小規模なシステムに対比して、会社の基幹業務に関わる大規模なシステムをさす。

基幹データベース[backbone database／enterprise database]

全社的な情報システムの中核をなすデータベース。外部に提供する商用データベースと異なり、自社内のユーザに限り使用できる場合が多い。自社の所有するデータベースのほか、自社内のユーザに提供する各種データベースを含む。(参)インハウスデータベース

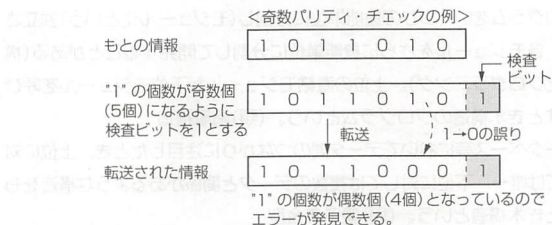
奇偶検査[parity check：PC／odd even check]

情報を転送する際、誤りなく情報が送られたかどうかを、コードの桁数字の和に注目してハードウェア的に検出するチェック方法のひとつ。情報のまとまりごとに1ビットの検査ビットを設け、そのビットを含め1であるビットの総数が常に偶数個(または奇数個)になるようして転送する。転送中に1ビットの誤りが発生するとビットの和の奇偶が合わなくなるため、エラーを検出できる。

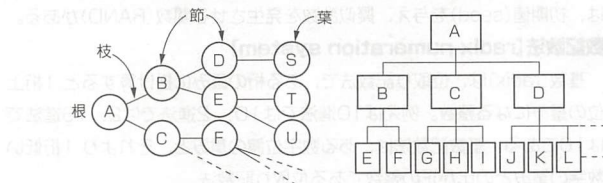
(1) 1であるビットの総和が常に偶数となるようにする方式を偶数パリティチェックといい、奇数となるようにする方式を奇数パリティチェックという。

(2) また2進数字が行列の形式になっているときに、行に対して行う奇偶検査を水平パリティチェック、列に対して行う検査を垂直パリティチェックという。

(3) 磁気テープなどの媒体に記憶した2進コードの検査の場合は、媒体の運動方向に対し平行な方向のビットについて行う検査を水平検査、垂

直な方向のビットについて行う検査を垂直検査という。

木構造[tree structure]

グラフ理論においては、環や閉路をもたないグラフを木(tree)と呼ぶ。情報科学においては、データなどの要素の上下関係を含めて議論することが多い。情報処理の分野では、構造化プログラミング、データベースのデータ構造、データの検索、ソート、コンパイラの構文解析などと関連する重要な概念である。

①点を連結する線が木の幹から枝が、枝から小枝が出るように分岐を繰り返して構成したグラフの形態をさす。上位に対しては唯一、下位に対しては複数のポインタがある構造を木構造という。上位の点を親(parent)、下位の点を子(child)と呼ぶこともある。木構造には、(1)ループがない、(2)階層構造に対応する、という特色がある。

頂点や辺の性格により、いくつかの型に分けられる。

(1)2分木(binary tree)

根(root)側から葉(leaf)への道をたどるとき、節から二つ以下の枝が出るもの。2進木ともいう。0の場合が葉になる。

(2)線形木(linear tree)

枝が1本しかないもので、配列やリストの線形構造に相当する。

(3)多分木(multiway tree)

各頂点から最大m個(m≧3)の枝が出る。2分木と区別される。

(4)順序木(ordered tree)

ひとつの頂点が子をいくつももち、子の間に順序があるもの。

(5)有向木(oriented tree)

枝に方向性をもたせ、意味を与えたもので矢印で表す。

(関)平衡木、B木、AVL木

②プログラムをいくつかの機能単位に分割し(モジュール化という)独立させ、各モジュールをさらに機能単位に分割して開発することがある(構造化プログラミング)。上位の直結モジュールが下位モジュールを呼び出すとき木構造のプログラムという。(類)階層構造

③データベース等においてデータ間のつながりに注目したとき、上位に対しては唯一、下位に対しては複数のデータと関係があるような構造をもつとき木構造という。(同)親子集合型

④UNIX系OSやMS-DOSのファイル管理システムでは複数のディレクトリ、またはフォルダを階層構造に配するが、親ディレクトリは唯一、サブディレクトリは複数を配することができる。treeコマンドで階層を表示することができる。

机上デバッグ[desk debugging]

作成したプログラムを実際にコンピュータでコンパイルし、データを入力してテストする前に、コーディングと流れ図をもとにプログラムの論理を目で見て検査し、誤りを検出して手直しすること。(関)デバッグ

擬似乱数[pseudo random numbers]

関数や手続きを利用して作った乱数。数学的に厳密な意味での乱数ではなく、乱数列を長くとると繰返しが現れることがある。プログラム言語には、初期値(seed)を与え、擬似乱数を発生させる関数(RAND)がある。

基数記数法[radix numeration system]

基数(radix)は、位取り記数法で、ある桁の重みに掛け算すると1桁上位の重みになる整数。例えば10進法では10、2進法では2、16進法では16である。基数記数法は、ある数字位置の重みと、それより1桁低い数字の重みとの比が正の整数である位取り記数法。

基数(radix)をrとし　$n=K_m \cdot r^m+K_{m-1} \cdot r^{m-1}+\cdots\cdots+K_1 \cdot r^1+K_0 \cdot r^0$

$(0 \leqq K_m、K_{m-1}、\cdots\cdots、K_1、K_0<r)$

の関係が成り立つとき、$K_m K_{m-1} \cdots\cdots K_1 K_0$は、数nをr進法で表現した数となる。(例) 2進法

10進法	2進法	8進法	16進法	10進法	2進法	8進法	16進法
0	0000	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

奇数パリティチェック[odd parity check]　(関)奇偶検査

機能キャラクタ[control character／functional character]

特定の文脈中に現れて、制御機能を開始、変更または停止する文字。印字制御や通信制御で使う特別の文字群。エスケープコード(制御文字)などがこれに当たる。EBCDICコードやASCIIコードなどの文字集合では、英字や数字などの文字以外に、伝送の制御や表示、印刷での書式の制御を行う文字を定めている。通信や印字の際には、プロトコルやインタフェースを正しく設定しておかないと、機能キャラクタの解釈が正しく行われず無意味なデータを授受することがある。(類)エスケープコード

機能設計[function design]

外部設計(概要設計)で、システムの機能をサブシステム単位に分割し、その内容と関連を明確にすること。

機能テスト[functional test]

システム、モジュール、プログラムのテストの際、内部構造より機能に重点をおいて行うテスト。仕様書に基づいて、システムが実現すべき機能をもれなくテストできるようなテストデータを用意して実施する。不必要な機能があってもテストでは検証できない欠点はあるが、ホワイトボックステストにくらべテストケースは少なくてすむ。モジュール(単体テスト)終了後の結合テストや総合テストで適用する場合が多い。

ギブソンミックス[Gibson mix]

主に科学技術計算でコンピュータの計算能力を評価するために、J.C.Gibsonが考案した数値のこと。計算処理を行う際の固定小数点加減算、浮動小数点加減算、ストア、判断などの個々の命令の、実際のプログラムでの使用頻度を調べ、対象のコンピュータの命令の実行速度をそれぞれに掛け合わせて合計して求める。小さい値になるほど処理速度の速いマシンといえる。1命令の平均実行時間を知るのに便利である。

命令の種類		重み	命令の種類	重み
データの移動 (記憶装置↔レジスタ)		31.2	分岐	16.6
			比較	3.8
加減算	固定小数点	6.1	演算(レジスタを用いない)	5.3
	浮動小数点	6.9	けた送り	4.4
乗　算	固定小数点	0.6	論理演算	1.6
	浮動小数点	3.8	指標レジスタ命令	18.0
除　算	固定小数点	0.2	合　計	100.0
	浮動小数点	1.5		

基本計画[basic plan／primitive planning]

システム開発の最初の段階で行う、システム概要と開発工程の計画。大きく分けて3つの要素からなる。

(1)システム化計画：現状を分析し、問題点を明らかにし、解決策をシス

キ

テム化計画として立案する。

(2) **プロジェクト実行計画**：システム化計画に沿って、概要を設計し、開発に要する資源(人・物・金・時間)を見積もる。

(3) **ユーザの要求定義**：システムへの要求を定義し、その機能、性能、運用要件、必要なハードウェア、ソフトウェアを明らかにする。

基本交換法[exchange sort／sort by exchange]

整列を行うための手法のひとつ。交換法ともいう。表の左の要素から順に隣合う要素を比較し、順番が逆のときは入れ換えを行うことを繰り返す。最右の要素が最大値になる。続いて比較の領域をひとつ狭め、同様の比較・交換を繰り返す。これを2つになるまで行う。簡明なアルゴリズムであるが計算量が多く、要素数が大きくなると実用に適しない。

(類)基本挿入法、バブルソート

基本制御構造

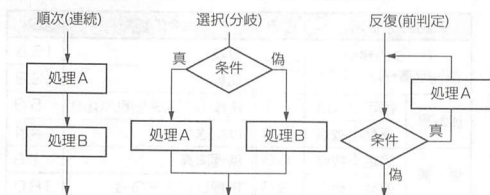
構造化定理で示すプログラムの基本の制御構造。さまざまな呼び方があるが、次の3つをさす。

(1) **順次形**(sequence 型)

プログラムの順に命令を実行する。連続構造、連接構造ともいう。

(2) **選択形**(IF THEN ELSE 型)

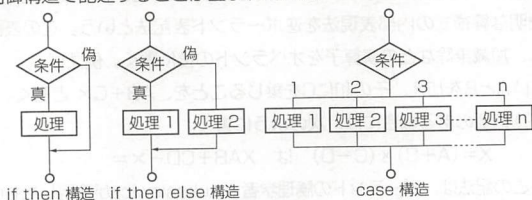
指定した条件により飛び越し先を変えて、いくつかの文の中から実行すべき文を選ぶ。二者択一により処理を実行する構造を if構造という。この指定を行う文、命令を分岐命令(branch instruction)、条件付き飛越し命令(conditional jump instruction)、条件判断命令(decision instruction)ともいう。条件の演算結果などにより条件コードの値をセットし、その値により飛越しを発生させる。

(3) **繰返し形**(反復構造)(DO WHILE 型)(関)繰返し

この他に、次のような2つの追加制御構造がある。

(4) **後判定型反復構造**(DO UNTIL 型)(関)繰返し

(5) **多分岐構造**(CASE 型)：三者以上の中からどれかひとつを選択して処理を実行する。case構造は if then else構造を多重化したものといえる。

論理が複雑で、長いプログラムも整理していけば、上記の構造で記述で

きる。GOTO文の多用はエラーの原因になりやすい、また開発者以外の人が読みとりにくいという欠点があるという意見(GOTO文有害論)が広く受け入れられ、GOTO文を極力使わず(GOTO LESS)に、プログラムを基本制御構造で記述することが望まれている。

基本設計[basic design]

コンピュータシステムの設計において、システムのおおまかな機能と構成を設計すること。予備設計ともいう。現状分析の結果に基づき、その問題点を明らかにしたうえで実現可能なシステムの基本構想を作成する。

基本挿入法[insert sort]

簡単な内部ソートのアルゴリズム。挿入ソートともいう。配列中のデータを整列ずみの部分と未整列の部分に分割し、未整列のデータを1件ずつ整列済みの列の中の正しい位置に挿入することの繰返しである。整列済みの配列については挿入位置の探索に2分探索法が使えるので効率がよい。

しかし、挿入のために配列中のデータの番地をずらすことが必要である。データをずらしながら挿入位置を探索するアルゴリズムもある。やはりデータ量の2乗に比例する計算量が必要であるが、データの大半が正順で、追加された一部データだけが乱順である場合には効率がよい。

基本ソフトウェア[basic software]

広義のオペレーティングシステム(OS)のこと。制御プログラム(狭義のOS、基本OS)、サービスプログラム、汎用言語プロセッサに分けられる。基本ソフトウェアとミドルウェアを合わせてシステムソフトウェアと呼ぶ。(類)オペレーティングシステム

基本データ領域[prime data area] (関)索引順次編成

機密管理[secret protection] (同)セキュリティ

機密性[security] (同)セキュリティ

逆アセンブル[deassemble/disassemble]

記憶領域の機械語命令コードを、それと1対1に対応しているアセンブラ形式の文に翻訳すること。通常はデバッグ時、デバッガのコマンドによって行う。なお、市販のソフトウェアは、著作権保護の観点から逆アセンブルを禁止している。(対)アセンブル

逆順[descending order] (同)降順

キ

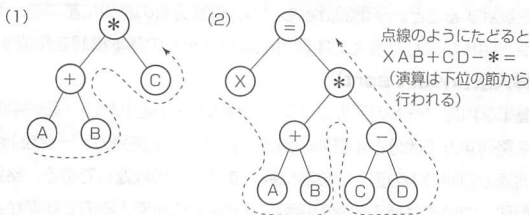
逆ポーランド表記法[reverse Polish notation]

FortranやCOBOLなどのソースプログラムをコンパイルする際に行う算術式を機械語に変換する過程では、加減算より乗除算を優先するなどの演算子の優先順位を考慮して、構文を明らかにする。スタックを利用した簡明な算術式の内部表現法を逆ポーランド表記法という。この表記法では、加減乗除などの演算子をオペランドの後に置く。例えば、

(1) AとBを加え、その和にCを乗じることを、AB+C× と書く。

(2) 括弧のある場合には、次のように書く。

X=(A+B)×(C−D)　は　XAB+CD−×=

この記法は、ポーランドの論理学者 J.Lukasiewiczが使用したもので、括弧なしで数式を表現できる。そのため、算術式の評価が単純に行える。後置表記法ともいう。(1)、(2)は構文木では図のように書けるものである。

キャッシュメモリ[cache memory／disk cache]

①主記憶装置の遅い動作速度と制御装置の速い動作速度の差を埋める(緩衝する)ために、両者の中間におく制御装置なみの速さをもつバッファ記憶装置。通常、数キロバイトから十数キロバイトの容量のバイポーラ型メモリである。CPUの高速化をはかり、キャッシュメモリ自体をCPUを構成するLSIの中に組み込んだものを、オンチップキャッシュ(on-chip cache／internal cache／primary cache)という。

②主記憶装置と磁気ディスク装置などのDASDの間に置いて、アクセス時間の差を調整する(緩衝する)ための緩衝記憶装置。特にディスクキャッシュメモリという。OSが、主記憶領域の一部をディスクキャッシュメモリとして割り当てることもできる。

キャラクタセット[character set]　(同)文字集合

キャリー[carry]

基数記数法の位取りに関する語で、加算を行う際の桁上げ、桁上がり。2つの数の加算を行って、ある桁で基数(2進数の2や16進数のF)に等しいか、基数を越えたときに、ひとつ上位の桁に1を加える処理。

行[row／line]

一般に、一連の文字列や数字列が横に並んだもの。情報処理では次のよ

うな例がある。

(1)数学用語では行列の行のこと。列(column)の対語。

(2)ソースプログラムのリストの1行。ソース行ともいう。

(参)行番号

(3)ワープロソフトなどのテキスト表示では、画面上のカーソルの上からの位置、文書のページの上からの位置。縦組表示の場合は、ページの左からの位置。

(4)表計算ソフト(スプレッドシート)やリレーショナルデータベースの関係表の場合、データを縦、横にグループ化し、横並びのデータを行とし、縦並びのデータを列という。(類)組、アトリビュート

(5)プログラムの二次元配列の場合、配列の大きさを宣言する先の変数。例えば、A(5、10)は 5行10列の大きさを表す。(対)列、桁

業界標準[de facto standard] (同)デファクトスタンダード

共通応用ソフトウェア[common application software]

アプリケーションプログラムのうち、比較的広い分野、多くの業種、業務で共通に使うソフトウェア。汎用性の高い機能を有しており、特定の業種、業務に特化しておらず、使用するときユーザの環境や要求に合わせるためにカスタマイズという作業が必要になる場合がある。

例えば、CAD／CAMのプログラムは、設計図なら巨大なビルでも、微小な部品でも描くことができ、設計の必要な業種であれば便利に使うことができる。また、自動翻訳のプログラムは、機械用とか医学用などの分野別の用語辞典などをオプションで追加することにより、広い分野で使用される。その他、統計処理ソフト、表計算ソフトなどのようにソフトウェア機能に汎用性がある種類を含む。

(参)個別応用ソフトウェア

行番号[line number]

ソースプログラムの各行につけた番号。Fortranでは文番号という。プログラムエディタ固有の行番号はプログラムリストなどに印字するが、コンパイラには影響を与えない。行番号は、かつてのFortranやBASICのプログラムでは、行番号が飛越し先や制御の範囲などを示す目印になり制御に関わっていた。

局所変数[local variable]

変数名の有効範囲が、それを定義したブロック内または関数内においてのみ有効な変数。局所変数は大域変数と違い、他のブロックと同一の変数名でも、定義したブロック範囲外では関係がなく、変数名を付ける場合の自由さが増す。局所変数に値を引き渡すためには引数などを利用する。例えば、Fortranのサブルーチン内の変数は局所変数である。

キロ[kilo：k／K]

10^3(＝1000)を意味する補助単位。ただし、情報処理の分野では、2^{10}(＝1024)がこれに近く、キロビットとは1024ビットのこと、キロバイトとは1024バイトのことがある。この場合、特にKbit、Kbyteと大文字で書き、「ケイビット」、「ケイバイト」と読むことがある。

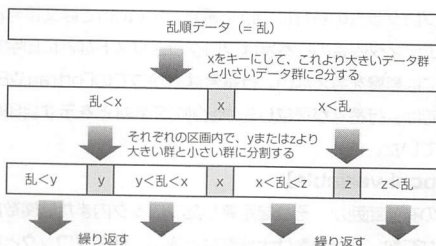
記録密度[packing density] (同)記憶密度

記録容量[recording capacity]

記憶媒体に記録できる情報量。フロッピーディスクなどでは、(1)本来の媒体が記録できる最大容量をさす場合と、(2)ユーザやOSが装置の仕様に合わせてフォーマットした後の媒体が記録できる容量をさす2つの場合がある。

ク

クイックソート[quick sort]

内部ソートのアルゴリズムの一つ。基本的な考えは、2分探索に似ている。配列中から1件のデータを選び、これより大きなデータを上位に、小さなデータを下位に移動する。データはある値を境に2分割されたわけである。大きいデータ群と小さいデータ群に対して別々に同様の操作を行えば、データは三つの値を境に4分割される。データ群ごとにこれをさらに細かく繰り返せばソートが完了する。分割交換整列法ともいう。平均計算量は、n件のデータついてnlog2nに比例、最悪の計算量はnの2乗に比例するが、実用上よいアルゴリズムと評価されている。なお、このアルゴリズムは改良され、さまざまな高速化アルゴリズムがくふうされている。

偶数パリティチェック[even parity check] (関)奇偶検査

空白文字[blank／space character]

情報交換用符号で、空白、間隙、余白を表す文字(文字定数の' ')。計算

機の内部または記憶媒体においては、空の桁ではなく文字コードがある。また、空白文字は、数字やアルファベット文字と同じように定められたビットパターンをもっている。コード表では「SP」で示す。間隔文字、スペース、ブランク、空字、また単に空白ともいう。

空文字列[null]

文字数ゼロの文字列。文字がひとつもない文字列。空白ではないことに注意。(同)ナル

クエリー[query]

データベース管理システムで、データベースから検索するデータを指定するルール。システムによって指定法は異なるが、指定のための言語があり、クエリー言語という。(関)SQL

区切り文字[separator/separate character/delimiter/punctuator]

文書やソースプログラム中の文字列(語)と文字列を区切るための1ないし2文字の記号。分離符、デリミタともいう。レコード中のデータを分離し、組み合わせる文字。多くのプログラム言語やアプリケーションでは、1文字では、空白文字、コンマ(、)、括弧、コロン(:)、セミコロン(;)、スラッシュ(/)、タブなどを用いる。2文字では、各種のかっこ、引用符などがある。(参)CSV

区点コード

漢字を表すコード体系のひとつ。1文字を指定するのに2桁の10進数の区番号と2桁の10進数の点番号で示す方式。
(対)シフトJISコード

区分①[segment]

データやプログラムを分割した単位のこと。(同)セグメント

区分②[partition]

記憶領域を分割した単位のこと。プログラムやデータをディスクに記録するとき、ディレクトリ(メンバ名と格納場所の情報をもつ)を設け、これによって構成メンバ(データやプログラム)を取り出せる方法を用いる(区分編成ファイル)。このとき構成メンバを入れる記憶場所を占有するが、これを区分という。構成メンバが増えると区分も増える。また、削除するとその区分が空地となる。空地に入れるのは同じメンバ名のファイルであるが、もとの大きさより大きいものを入れることはできない。
(参)区分編成ファイル

区分③[block]

データがその属性や内容により分類できるとき、分類のクラスを区分ということがある。(参)区分コード

区分化[segmentation]

1つのプログラムをいくつかの単位に分割すること。主記憶装置の容量の割に大きなプログラムを作る場合に行われる。

区分コード[block code]

コード化の対象をいくつかのブロック(グループ)に分けるとき、ブロック内で連番をつけたコードで、項目を特定の意味のあるブロックに分類したもの(市町村コードなど)。ブロック別コードともいう。

桁数が少ないうえ、空番(欠番＝予備コード)があるためある程度の追加は可能で、グループ集計も可能で分類に便利である。しかし、追加やデータが多くなると不便である。

〈例〉大学・高等専門学校コード

0001～0999 国立大学、　1001～1999 公立大学、

2001～2999 私立大学、

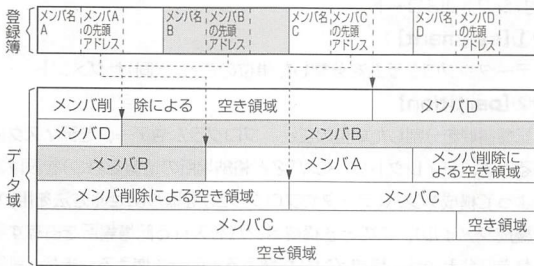
3001～3999 国立短期大学、　4001～4999 公立短期大学、

5001～5999 私立短期大学、

6001～6999 国立高等専門学校　等。

区分編成ファイル[partitioned organization file]

順次編成ファイルをメンバという単位のサブファイルに分割し、別に各メンバの開始位置を示す登録簿を作ったファイル。登録簿のアドレスを見てからデータ域のメンバごとにアクセスする。メンバは順次編成となっているため、分割形順次編成ファイルということがある。ライブラリ形式のファイルは、ほとんどがこの編成法をとっている。

組[tuple]

通常は1件分のデータ。カード型データベースの場合はカード1枚に相当する。リレーショナルデータベースの二次元表の行のこと。アトリビュートが示すデータの組で、ファイルのレコードに相当する。

組込み関数[intrinsic function／built-in function]

FortranやBasicの用語で、三角関数、平方根、整数化などプログラム

中でよく使う関数をあらかじめコンパイラに組み込んでおき、プログラマがソースプログラムを記述するときに呼び出して引用できるようにした関数。文関数や外部関数が定義をしてから引用するのに対比して、組込み関数という。

クライアントサーバシステム[client server system：CSS]

分散処理システムの一形態。処理を要求するクライアントと、クライアントに各種サービスを提供する専用コンピュータ(サーバ)とをネットワークで接続した情報処理システム。LAN(構内通信網)などのネットワーク上に複数のクライアントとサーバを置くことも可能で、機能を分担して負荷分散を図ることができる。水平分散処理システムの代表的形態である。クライアントサーバシステムでは、接続しているコンピュータ間に上下関係はなく、相互に受持ちのサービスを提供する役割に変わっている。

これに対し、従来のホストコンピュータ中心の分散処理では、端末は単に入出力機器として扱っており垂直分散処理と呼ぶ。クライアントサーバシステム実現の背景には、高度な機能をもつパソコンやワークステーションなどが出現したことがある。(参)リモートジョブエントリ

クラック[crack]

アクセス権のない者がシステム内部に不正に侵入すること。侵入者のことをクラッカーという。類語のハッカーは必ずしも悪い意味で使うとは限らないが、クラッカーは犯罪者に近い意味で使う。クラックの手口は、他人のアカウント(利用者IDやパスワード)を盗み、プログラムの欠陥や運用上の管理不徹底などのセキュリティホールをついて侵入する。

(類)ハッカー

クラッシュ[crash]

① 磁気ヘッドの破壊。ヘッドクラッシュともいう。

② パソコンシステムの障害で、システムがコマンドを受け付けない状態になること。ハングアップともいう。

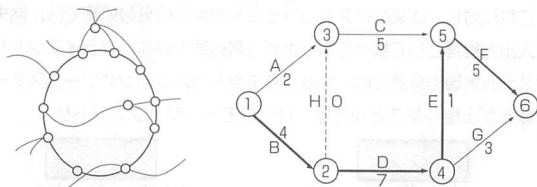
グラフ[graph]

①数字や文字を配列した表に対し、統計量を図形的な要素(長さ、角度、面積)で表し、変化・比較・対照などを行う図。表現する内容により、各種のグラフがあり、統計量の要点・変化などを一目で把握できる利点がある。表計算ソフトやプレゼンテーションソフトには、数値の列を各

種グラフにして表示する機能があり、これをグラフ機能という。

②データ構造でいうグラフは、距離や長さを無視して、いくつかの点と隣接関係にある点を結ぶ線で表す図。さまざまな事象の間の関係を、事象を点で、事象間のつながりを線で表現したのがグラフである。線に方向をもたせたり、重みをもたせたりすることもできる。例えば、駅を点で表し、所要時間や距離、料金などを線で表す路線地図はグラフである。

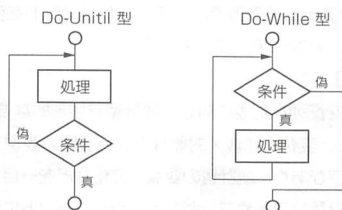
点を頂点(vertex)、または節点(node)ともいい、結ぶ線を辺(edge)、孤(arc)、または枝(branch)ともいう。ある点から別の点に結ぶ一連の枝があるとき、この経路を道(path)という。ある頂点から別の頂点を経由して自分自身に帰る道を環、または閉路(cycle/closed path)という。

(参)木構造、ネットワーク、状態遷移図

クリアする[to clear]

記憶装置内の記憶場所や変数、配列の値を、ある決められた状態に戻すこと。通常ゼロや空白文字を格納する場合が多い。

(参)ゼロクリア、初期化

繰返し[repetitive/iteration]

基本制御構造のひとつ。同一文(statement)を一定の条件下で何度も反復実行すること、その制御文。反復ともいう。繰返しの回数を制御するため、回数の指定や、ループの脱出条件などの設定を行う。通常、表(table)操作をともなうことが多く、繰返しを制御する変数を添字として使用し、簡潔な文で複数の命令を実行したのと同一の効果が得られる。繰返し処理の内部に、別の繰返し処理を組み込んで入れ子の構造で利用することができる。

Do Until型とDo While型があり、前者はどんな場合でも必ず一度は処理を実行するが、後者は条件によって、一度も実行しない場合がある。

高水準言語では、これに対応する各種制御構造が整っている。COBOL言語ではPERFORM文、FortranではDO文、BASICではFOR…NEXT文、REPEAT…UNTIL文などが繰返しを表現する命令である。

グループインディケーション[group indication]

グループ化されたデータを印字するとき、グループ名の変わり目や行頭のみに印字し、同名のグループ名は印字を省くこと。例えば、クラスごとに整列してあるデータから成績一覧表を作成するとき、最初の1行だけクラス名を印字し、その後はクラスが変わるまで空白に置き換えると見やすくなる。(参)グループトータル

グループウェア[groupware]

ネットワークを使用して、オフィスでの共同作業を総合的かつ円滑に行うためのソフトウェア。おもな機能は、電子メール、電子掲示板、会議のスケジュール管理、住所録、文書管理、電子会議システムなどである。従来、電話や文書でやり取りしていた情報交換を端末の前にいながらにして行えるようにすることを目的にしている。具体的には、グループ内の仕事の流れ(ワークフロー)を支援するグループウェア機能を活用する。グループウェアを使用して共同作業を行うことをCSCWという。なお、決済や契約を行うために印鑑捺印処理や認証を行うシステムも登場している。

グループトータル[group total]

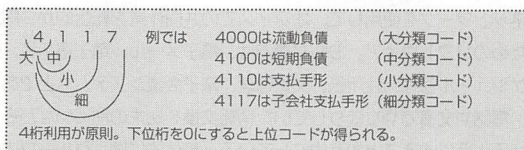
あるキー項目について整列したデータをそのキーの値ごとに集計する処理。例えば、部課単位や商品単位などのように部門別の集計をするような場合、ひとつのグループとして同じ性質のものを集計し、さらにすべてのグループの合計を求めるような処理のこと。例としては、売上一覧表にお

キー		データ	トータル		
中コード	小コード	データ			
		データ			
		データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	
中コード	小コード	データ			
		データ			
		データ	小計		
	小コード	データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	
	小コード	データ			
中コード		データ	小計		
中コード	小コード				
		データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	大計

ける商品別売上金額の大計、中計、小計を求めることに相当する。

(参)コントロールブレイク、グループインディケーション

グループ分類コード[group classification code]

分類コードの一種で、階層構造のデータを所属するグループごとに分類するコード。コード化対象データを、大分類、中分類、小分類などに分類し、コードの各桁に対応させ、各分類内で一連番号をつける。桁別分類コード、または単に分類コードともいう。データが多くなると桁数も多くなる。例えば、会社などの組織のコードを部、課、係などに分けて、それぞれに一連番号を付けて用いる場合など。各桁の数字が1グループに対応し特定の意味をもつ。下位の桁を切捨てるか、0に置き換えると上位のコードを得る10進分類コードが代表的。JISには、産業分類コード、職業分類コード、勘定科目コードなどがある。下の例はJISの勘定科目コード。

(参)分類コード、10進分類コード

グレイコード[Gray code]

整数を表現するための2進表記法の一種。連続する2つの数の数表示がひとつの数字位置だけ異なる。例えば、数0から9の数表示は、0000、0001、0011、0010、0110、0111、0101、0100、1100、1101で1桁の交番コードを表している。またアナログからディジタル変換のときに誤差が小さく、入出力デバイスに広く使用する。

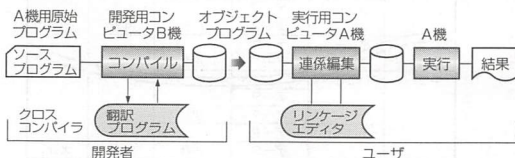
クローズドシステム[closed system] (対)オープンシステム

クローズドバッチ処理[closed batch processing]

コンピュータの操作を専任のオペレータに任せるバッチ処理方式で、ユーザは処理したいデータやプログラム、処理手順を一括してオペレータに渡して処理を依頼、処理後のデータやプログラムを受け取る運用の方式。

クロスアセンブル[cross assemble] (関)アセンブル

クロスコンパイラ[cross compiler]

他の計算機システムで実行するためのオブジェクトモジュールを生成

するプログラム。実行させるのとは異なる計算機でコンパイルが可能なため、プログラム開発が効率化できる。特にクロスコンパイラと区別して、自機種用コンパイラをさすときにはセルフコンパイラという。

群計数チェック[group check／block check]

情報伝送の誤り検出のためにブロック内のビット1の数の和をとり、その下位 2ビットを情報に付加するチェック方式。

ケ

計算量[complexity]

アルゴリズムの評価基準のひとつ。答えを得るまでの計算に要する実行命令数。同じ答えを得るなら計算量の少ないアルゴリズムのほうがよい。計算の複雑さ、手間ともいう。(参)時間計算量、O記法

継続行[continuation line]

ソースプログラムでステートメントが1行に書ききれないとき、引き続き文字を記入するための次の行。Fortranでは、6けた目に0(ゼロ)以外の文字、COBOLでは、7けた目に'-'を記入すると継続行として扱う。

経歴ファイル[history file／historical file] (関)更新履歴

桁[column／position]

①数を表す記数法で、位取りを表す位置のこと。(参)基数記数法

②ディスプレイ画面やプリンタ用紙に出力される文字の位置を表す単位。左端から何字目かを表す数字のこと。半角の文字を基準に数える。全角文字は半分の文字数になる。

桁上げ／桁上がり[carry] (同)キャリー

桁あふれ[overflow] (同)オーバフロー

桁送り[shift] (同)シフト

桁落ち[underflow] (同)アンダフロー

桁落ち誤差[cancellation error]

ほとんど近い2つの数値の差を計算する場合、上位の桁の相殺によって有効数字が減少してしまう。この誤差をいう。例えば、$\sqrt{200}-\sqrt{199}$を求める場合、直接なら、

$$\begin{array}{rll} \sqrt{200} & \rightarrow \ \ 14.1421356237 & (12桁) \\ -\sqrt{199} & \rightarrow -14.1067359796 & (12桁) \\ \hline & \ \ \ \ \ \ 0.0353996441 & (\ 9桁) \end{array}$$

この例の場合は分母の有理化によって改善する。分母、分子に

$(\sqrt{200}+\sqrt{199})$を乗じると、$(200-199)/(\sqrt{200}+\sqrt{199})=1/28.2488716033=0.03539964406$(10桁)

となり、やや改善される。

桁借り[borrow] (同)ボロー

結合[join] (参)関係代数

結合テスト[combined test/joint test]

関連する 2つ以上のプログラムモジュールを連結(結合)して、正しく動作するかどうかをテストすること。個々のプログラムが仕様書の機能どおりに動作するかを調べるモジュールテスト(単体テスト)で正常に動作しても、結合してみてはじめてエラーを発見するケースがあるので、モジュールの結合順序や結合時期を考慮しながら行う必要がある。

決定表[decision table] (同)デシジョンテーブル

言語処理プログラム[language processor] (同)言語プロセッサ

言語プロセッサ[language processor]

プログラム言語で記述したプログラムの翻訳や解釈、変換などを行うプログラム。言語処理プログラム、翻訳プログラムともいう。ある国の言語を別の国の言語に、あるプログラム言語を別のプログラム言語に翻訳する。(類)翻訳プログラム、(例)コンパイラ、インタプリタ、ジェネレータ、アセンブラ

検査①[check] (類)チェックシステム

検査②[test] (同)テスト

検索①[retrieve]

データベースの中から必要な情報を体系的に整理・分類しながら引き出すこと。検索項目のキーに種々の条件指定をつけることで、特定のデータを迅速、かつ正確に再編成、あるいは表示することができる。例えば、商用データベースの情報検索システムでは、キーワードの記述、条件の指定などを対話的に行える専用のコマンド言語がある。

検索②[search/find]

コンピュータシステムのファイル管理機能、データベースソフト、ワープロソフトの文字列探索機能のひとつ。対象とする装置、ディレクトリ(フォルダ)、ファイルから指定した文字列を探し表示すること。ワイルドカードや正規表現を使ったり、条件式を使ったりして検索できるソフトが増えている。

検査プログラム[check program]

①システム開発では、コンピュータが正しく働くかどうか、またソフトウェアが間違いなく作られているかどうかを確かめるプログラム。(参)ダンプ、トレース

②ユーザの分野では、コンピュータシステムのメモリ、ディスクの領域が正しいどうか、またソフトウェアにウィルスが潜伏していないか、文書のスペルに間違いがないかなどを確かめるためのプログラムの総称。

検査文字[check character] (同)チェックコード

原始データ[source data] (同)ソースデータ

原始プログラム[source program] (同)ソースプログラム

検収[acceptance]

システムの開発・テストが終了し、開発部門から利用・運用部門に移行できると判断すること、または運用部門がシステムを受け入れること。

現状分析[system analysis/status quo analysis]

システム開発に先だち、現行のシステムで情報がどのように処理されているかを知ることを目的として行う調査。これによって、現行のシステムにどのような問題点があるかを具体的に知ることができ、またいろいろな業務活動についても知ることができる。

件数検査[count check]

入力データの重複やもれをチェックするために、原始データの件数を調べる方法。伝票の枚数などを人間が数えて入力し、いっぽう、システムも入力データの件数を数えて、双方が一致するかどうかを確かめる。

限度検査[limit check] (同)リミットチェック

原票設計[source document design] (同)帳票設計

コ

語[word] (同)ワード

子[child/member]

①木構造において上位の節を親と呼ぶときの、下位の節。特に順序木(ordered tree)は、ひとつの節が子をいくつももち、その子の間に順序関係がある。

②親子集合型(set type)のデータベースで、相対的に上位のレコードを親レコード、下位のものを子レコードという。階層型の代表的なDBMSであるIMS(IBM)では、親をペアレント(parent)、子をチャイルド(child)という。また、CODASYLのDBTGでは、親をオーナ(owner)、子をメンバ(member)という。

項①[item] (同)項目、(参)データ項目

項②[term]

演算の対象となる変数やオブジェクトのこと。例えば、単項演算子は、

1つの項をもつ。(同)オペランド

広域変数[global variable]

Cの用語で、広域変数とは、プログラムの実行中に存続し、任意の関数から値が参照可能な変数のことをいう。広域変数は、関数の外で宣言することにより作成される。変数の宣言は変数の使用前ならいつでもよいが、プログラムの当初で宣言するのが確実である。

(対)局所変数、(同)大域変数

公開かぎ暗号系[public-key crypto system]

暗号化方式の一種。暗号化かぎと復号かぎが別で、暗号化かぎのほうは公開する方式。暗号化はだれでもできるが、復号かぎを知る人しか復号できない。この方式は、閉めることはだれでもできるが、開けるにはかぎが必要な金庫に例えることができる。不特定多数の人が限られた相手(審査員など)以外に見られないよう送信(守秘通信)したいとき便利である。また電子商取引(EC)のように確かに本人であることを証明(認証)するために利用できる。さらに通信文が改ざんされていないことの確認、発信者による発信行為の否認を防止できる。ただし、かぎの長さが慣用暗号系(共通かぎ方式)に比べ10倍近く必要で、暗号化速度、復号速度も遅いのが欠点である。代表的な方式にRSA暗号、エルガマル暗号、クラマー・シャウプ暗号がある。

(例)RSA、PEM、PGP、(関)ディジタル署名、認証

属性＼種別	慣用暗号系	公開かぎ暗号系
暗号化かぎ複合かぎの関係	同一	異なる
暗号化かぎ	秘密	公開
復号かぎ	秘密	秘密
かぎの大きさの例	64ビット	512ビット
暗号化速度	速い	遅い
秘密のかぎの受信	必要	不要

高級言語[high level language]　(同)高水準言語

合計検査[batch total check／total check／sum check]

入力に際し、データとその合計(手計算より得た値)を入力し、計算機側でも入力した値により別途合計を計算し、照合することで、入力の誤りやデータのもれ、過不足などを検出する検査法。(参)ハッシュ合計

降順[descending order]

データのソート処理で、配列やテーブルなどの特定のキー項目について大きい順から小さい順へ並ぶ順序。昇順の反対の順序。逆順ともいう。文字列の場合、ZからAへの順序になる。(対)昇順、(参)ソート

更新[update]

すでにあるファイル中のレコードの追加や削除、変更などの処理。アップデートともいう。順編成ファイルではファイルを読み込み、更新対象となるレコード内容を書き換え、新たにファイルに出力する。索引順編成ファイルでも、順次アクセスの場合は順編成ファイルと同様、直接アクセスの場合はキーにより、レコードを読み込みレコード内容を書き換える。

更新履歴[update history]

更新は、ファイル中のレコードの追加や削除、レコード中の項目の変更などを行うこと。アップデイトともいう。履歴は、システム稼働中、どのような処理をどの使用者が行い、処理がどのように終了したかなどを記録しておくこと。記録するファイルを履歴(経歴)ファイル(historical file)という。更新履歴の保存は、誤った処理、ファイルの追加や脱落などの事故にそなえる措置である。プログラムやデータなどのマスターファイルを更新する場合、更新後も前回、前々回のファイルを定めた世代(または定めた期間)にわたって保存するのが通例であり、こうしたファイルの管理法を世代交替法(generation system)という。

高水準言語[high level language]

汎用プログラム言語を大きく「機械向き言語」と「問題向き言語」に分けたときの問題向き言語。機械向き言語と違い、プログラミングの容易な言語で、主としてC、C++、Pascal、COBOL、Fortranといったコンパイラ言語をさす。機械に密接した言語ではなく、ハードウェアからの独立性の高い、より人間の思考に近い言語と言える。逆に機械向き言語は機械語やアセンブラ言語のことをさし、機械に密接した言語で、低水準言語という。(参)問題向き言語、(対)機械向き言語

構成管理[configuration management]

システムやネットワークを構成するハード、ソフトの機能、性能の維持、正常な運用の維持のために資源を管理すること。

合成コード[synthetic code]

コード化の一技法で 2つ以上の異なるコード体系を組み合わせてコードとしたもの。表意コードとの組合せが一般に多い。例えば、学籍番号を表す場合、学部(表意コード)+入学年度(表意コード)+クラス(表意コード)+出席番号(順番コード)などの複数のコード体系を用いる。多目的利用は可能であるが、桁数が多くなりがちである。

構造化設計[structured design]

大規模なソフトウェアを誤りなく効率的に開発するプログラミング規範として、また既存のプログラムの保守に多くの時間を費す事態に対処するため、プログラムの内容を把握しやすくするために考案したプログラムの

設計方法。大規模なプログラム開発時には、プログラムの処理機能を分割し、モジュール化する。各モジュールをさらに機能単位に分割するなどして階層構造をもたせる。このようなモジュール構成でプログラミングする手法を構造化プログラミングという。
(関)モジュール、階層構造、段階的詳細化、トップダウンプログラミング

構造化チャート[structured chart]

アルゴリズムを図式的に表現する図法。流れ図の矢印(プログラム言語ではgoto文)が不要であることが望ましいとした構造化定理に基づいた図法である。

構造化定理[structure theorem]

計算機科学者C.BohmとG.Jacopiniが 1966年に発表したプログラムの構造原理。入り口ひとつ、出口ひとつのプログラムは 3つの制御構造の組合せですべて表現できることを示した。(関)基本制御構造

構造化プログラミング[structured programming：SP]

① 大規模なソフトウェア開発においては、構造化設計と同義。
(同)構造化設計

② 小規模なプログラム開発、またはモジュールやサブプログラムの開発においても、プログラムの機能を分割し、制御構造を明らかにするのが望ましい。

③ 構造化コーディング(structured coding)は、手続き型言語を使用して、3つの構造で論理を組み立て、GOTO文を使わずにプログラムを作成すること。基本制御構造のプログラムは、個人差が少なくなるために読みやすく、構造が見やすく整理できているために、検証しやすくなり、結果的に信頼性の高いソフトを作ることができる。GO TO文が多いプログラムは、論理が複雑でエラーの原因になりやすい、また開発者以外の人が読みとりにくいという欠点があった。(参)基本制御構造

構造体[structure]

データ型の異なる複数の値で構成したデータ構造。レコード型ともいう。1つの名称で参照できる複数の変数の集合体であり、個々の構造体の要素(メンバ)はメモリ上の連続した領域に確保される。

(1) COBOLではレコードとして扱い、このような構造をもつ配列を定義することができる。
(2) Fortranでは、利用者定義データ型として新たに定められた。
(3) Cでは、複数の変数をひとまとめにして扱う構造体名で扱える。構造体の要素として、再帰的に自分自身を含むことはできないが、自分自身へのポインタを含むことはできる。

構文[syntax]

プログラムの形式的な構造に関する文法規則。特にコンパイラ言語などの高水準言語では、ソースプログラムを正しく構成するための文法を厳密に規定してあり、誤った文は翻訳することができない。そこで、コンパイラの中には必ず字句や構文チェック(syntax check)の機能がある。コンパイラは、コンパイル中に翻訳不可能な誤りを検出すると、誤りの場所、種類などを出力する。これをコンパイルエラーメッセージという。

構文解析[syntax analysis／parsing]

①機械翻訳や自然言語処理で、入力した文がどのような文法構造になっているかを分析すること。解析法としては下向き法や上向き法などがあり、前者の例はLL(k)法が、後者の例はLR(k)法などである。構文解析プログラムのことをparserという。

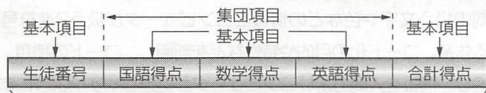
②ソースプログラムのコンパイル時、コンパイラは、ソースプログラムを分析する。英数字などの識別子、定数、演算子などの構文を構成している要素に分解した後(字句解析)、それらがどのように組み合わさっているかを調べ(構文解析)、オブジェクトプログラムを生成する。コンパイラは通常構文解析と同時に、あるいはその後、意味解析も行う。(参)意味解析、字句解析

構文木[syntactic tree]

ソースプログラムの文や句の構造を木構造で表現したもので、木の各節点には文法的なカテゴリ名(例えば、演算子名)が与えられ、葉には単語名(例えば、変数名や定数名)が与えられ、根にはその字句列全体のカテゴリ名(例えば、代入命令)が割り付けられる。根を上に、葉を下に書くのが通例である。代表的な構文木のひとつに算術式の構造を表す逆ポーランド表記法がある。

項目[item]

①広義には、データ集合の構成要素のこと。単位として取り扱われる一連の文字列、または語の意味の集まり。

②レコードを構成データに与える意味のこと。ファイルはレコードのようないくつかの項目からなり、さらにそのレコードは他の項目からなる。それ以上細分化できない項目は基本項目といい、基本項目が集まってできた要素は集団項目という。例えば、日付データは年・月・日の3つの項目の集合で、個人データは氏名、生年月日、性別、住所などの項目からなる。

(類)変数、フィールド、レコード、ファイル

コーディング[coding]

①流れ図などで表した論理的な手順にしたがって、プログラム言語を用いてプログラムを記述すること。言語がコンパイラ言語であってもアセンブラ言語であってもコーディングという。(類)プログラミング

②EDPの対象となるデータの項目にコード化をすること。ファイル設計の重要な要素となる。(同)コード設計

コーディングシート[coding sheet]

原始プログラムを作成する際に用いる用紙で、プログラム言語の書式固有の形式を印刷したもの。プログラム開発に必要な諸情報を記入する欄を設けてあるのが普通である。コーディング用紙ともいう。

コード[code]

①(情報交換用符号)文字データの表現方法を規定するあいまいさのない規則や体系。(参)コード体系

②(データコード)データを識別するために、使用目的に応じて、識別、分類、あるいは配列するために、事象名につけた記号や略号。
(参)コード設計

③(プログラムのコード)データの所在や命令の種類をある形式に変換したもの。

コード設計[code design]

データ処理に際して、データを識別、分類、区分、照合、あるいは配列を容易化、効率化するために、データに一定の基準を設け体系化を図ること。コード設計でいうコードとは、データコードのことで、使用目的やデータ量に応じて、事象名につけた符号である。この対応づけをコード化という。通常、次のような順で行う。

(1)処理すべきデータの選定(コード化の対象など)
(2)データの体系化(識別・分類・配列)
(3)コード化の調整と決定(操作性・表示のわかりやすさ・エラー検出など)
(4)コードの文書化
(5)コードの管理

コード化の方式については、順番コード、区分コード、分類コードなどがある。

コード体系[coding scheme]

人間の扱う文字や色などの情報とコンピュータが扱う2進符号列を対応させる体系。コード化の目的や特性などを判断し、コードの種類、桁数などを規定し、情報を人間およびコンピュータが処理しやすいようにまとめる。

①情報を符号で表現する規則。例えば、JIS 7単位符号、EBCDIC、シフ

トJISコード、EUCコード、UCSコードなどがある。

(参)コード、情報交換用符号

②情報の送信・受信にともなう符号化方法の規約。

コード入力[input by codes／code input]

①コード化した情報を、そのコードで入力すること。例えば、「企画部」を00、「総務部」を01としたとき、「企画部」と入力する代わりに00と入力すること。

②日本語の文字入力をするとき、FEPなどの辞書機能を使わず、漢字コード(JISの区点コード、シフトJISコード)を直接入力すること。特殊な記号や文字、辞書に登録していない文字、外字の入力に使用する。

コード変換[code conversion]

あるコード体系のコードを別のコード体系の同じ意味のコードに変換すること。あるシステムが、別のシステムが作ったデータを利用するとき、データの体系が異なれば、処理のためにデータ形式の変換が必要である。データやプログラムなどを共用するために必要な処理である。

コールドスタンバイ方式[hot stand-by system]

2組のシステムを用意し、一方のシステムの障害発生時に、別のシステムで対応するシステム構成。信頼性が厳しく要求されるオンラインシステムでは、デュプレックス構成にしてシステムの障害時にも稼働し続けるようにする。しかし、即時に切替えるデュアル構成システム(ホットスタンバイ方式)に比べ、主系ダウン時の切り替えに時間がかかることから、コールドスタンバイ方式という。(対)ホットスタンバイ方式

コール命令[call instruction] (同)呼出し命令

互換性[compatibility]

あるハードウェア、もしくはソフトウェアを別のものと置き換えたとき、オリジナルのものと同等に機能すること、または、その度合を示す。(関)移植性

誤差[error]

測定・観測した値と真の値との差異。情報処理の分野では計算した値と真の値との差を誤差という。計算値などから真値を代数的に引いた結果を絶対誤差(absolute error)といい、絶対誤差と真値との比を相対誤差(relative error)という。計算誤差の原因には次のようなものがある。

(1)データ自体の誤り。入力ミスや10進数を2進数に基数変換する際に生じる誤差。

(2)無限回の計算を有限回で打ち切ったり、アンダフローした値をゼロとする際に生じる誤差。(同)打切り誤差

(3)浮動小数点数を有限桁計算で行うための誤差。

(同)情報落ち誤差、桁落ち誤差

(4)整数演算、実数演算、または単精度、倍精度などの混用による丸めの誤差。(同)丸め誤差

故障[failure] (類)障害回復、システムダウン

語長[word length]

ワード(語)の長さ(ビット数)のことで、中央処理装置内でひとまとめにして転送されるデータの大きさの単位。一般にハードウェアにより決まり、固定語長という。過去にはこれを変えられる機種もあり、可変語長という。(参)ワード

固定長レコード[fixed length record] (関)レコードタイプ

コピー[copy]

①原典(オリジナル)のデータはそのまま残して原典からデータを読み取り、ほかの場所へ書き込むこと。書き込んだデータの物理的な形は原典での形と異なっていてもよい。事故や記憶媒体の劣化に備えるために行う。

(参)バックアップ、違法コピー、ミラーリング

②文書処理では、指定した文字列や画像を新たな位置に転写する機能。

個別応用ソフトウェア[individual application software]

アプリケーションプログラムのうち、利用者が自分の業務のために作成したプログラム。または特定の業種向けに、コンピュータメーカやソフトウェア会社が作成し、有料で提供するプログラム。適用業務プログラムともいう。各業務に固有の機能をもち、業種や業務が異なれば、利用できない、利用しにくい点で共通応用ソフトウェアと異なる。

(関)共通応用ソフトウェア

コマンド[command]

広義には、命令、指令のこと。またその信号を含む。ユーザが直接コンピュータシステムやアプリケーションに与える命令。主としてキーボードから入力したり、バッチプログラムに記述する命令をさす。

コマンド方式[command mode]

商用データベースシステムの利用形態。利用者が通信回線を介してDBMSに働きかけ、対話的にデータベースのデータを操作する方式。例えば、情報検索システムではキーワードの記述、条件の指定法など専用のコマンド言語がある。

コメント[comment]

文書やプログラム中に記述する補助的なメモ書き。ソースプログラムの場合、コンパイルの対象とならない。文書の場合は特に指定すれば印字されない。(同)注釈

コラボレーション[collaboration system：CS]

もとは科学技術ネットワークの用語で、未解決の大きな問題を研究するために多数の研究者が協同して利用するデータベースやコンピュータ資源をさす。転じて、同様の協調的なグループの作業を支援するシステムをさす。共有データベースには参加者が自分の実験結果や考察、意見などを書き込みメンバに公開し、討論の場ともなる。

コンカレント処理[concurrent processing]

並行処理。ある時間間隔内に複数の動作を同時に行う処理形態。CPUが、処理の遅いプリンタなどの入出力装置の動作終了を待たずに次の命令を実行することにより、CPUの遊休時間を減らし、システムを効率よく利用する。マルチプロセッサシステムのように複数のCPUで同時に複数の命令を実行する処理形態(並列処理)も含めてコンカレント処理という場合もある。(類)並行処理

コンディションコード[condition code]　(同)条件コード

コントロールカウンタ[control counter]　(同)命令アドレスレジスタ

コントロールコード[control code]　(同)機能キャラクタ

コントロールブレイク[control break]

整列ずみのデータの項目や内容が変化した状態をとらえて処理手順を変えるプログラミング技法。例えば成績処理では、各レコードを順に処理し、クラス、学年などのキー項目に変化が生じた時点(切れめ＝ブレイクポイント)を判定し、クラス別合計や学年別合計を印字するような処理をいう。(関)グループトータル、ブレイクポイント

コンバージョン[conversion]

変換すること。移植や翻訳と異なり、主として物理的、形式的な変換をいう。

①プログラム変換(program conversion)：あるシステム(A機)のソースプログラムを別のシステム(B機)でも利用したいとき、A機のソースプログラムをB機に適合するソースプログラムに変換すること。一般的には、ソースプログラムレベルでコンバージョンプログラム(プリプロセッサ、プリコンパイラ)を用いて変換し、コンパイルし直すことが多い。

②メディア変換(medium conversion)：フロッピーディスクからハードディスクにというような記憶媒体の変換をすること。

③コード変換(code conversion)：入力したデータと処理可能なデータのコード体系が異なる場合、処理のために入力データを処理可能なデータ形式へ変換すること。特にある符号体系から別の符号体系に変換すること。なお、②、③の変換をファイル変換ともいう。

④信号変換(signal conversion):電気信号の形式を変換すること。例えば、アナログ信号をディジタル信号に変換したり、直列信号を並列信号に変換すること。

⑤ 型変換(type conversion):データの型を変換すること。例えば、整数型の変数に実数型の演算結果を代入する場合などに必要な変換。

(参)暗黙の型変換、型

コンバータ[converter]

コンバージョンを行うハードウェアやソフトウェア。

①記憶媒体の変換には、ハードウェアによる物理的コンバータを使う。

②オペレーティングシステムやアプリケーションによって記録の方法が異なるデータの変換には、ソフトウェアによる論理的コンバータを用いる。異なる日本語ワープロ間のデータを相互に利用する場合や、日本語ワープロで作成した文書データをデータベースなどで読み込んで利用する場合なども同様である。

③電気信号の形式を変換する機器。アナログ信号をディジタル信号に変換したり、その逆の変換を行う機器などがある。

コンパイラ[compiler]

Fortran、COBOL、Pascal、Cなどの高水準言語で書いてあるソースプログラムを機械語に翻訳するプログラム。コンパイラは、それぞれの機械に固有のものを開発し、メーカ、あるいは言語開発者がユーザに提供するのが普通である。その種類や機能は多種多様で、さまざまな分類ができる。

(1) プログラムの実行に必要なサブルーチンや副プログラムをまとめて、一度にコンパイルするものを一括コンパイラという。これに対し、大きなプログラムをモジュール単位に分けてコンパイルし、後からリンクするものを分割コンパイラという。前者はコンパイル時にプログラム間の不整合を検出できる良さがあり、後者はあるモジュールをプログラムのライブラリとして活用できる良さがある。
前者の例にPascalコンパイラ、Cコンパイラ、C++コンパイラ、後者の例にFortranコンパイラやCOBOLコンパイラがある。

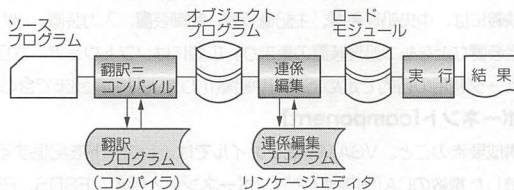
(2) 実行用コンピュータとは異なる機種のコンピュータ上で、実行用コンピュータの機械語コードを生成するものをクロスコンパイラという。これに対し、自機種で実行する機械語コードを生成するものをセルフコンパイラという。

(3) コンパイルの過程で、中間的なコードを何回か生成し、最後にオブジェクトプログラムを生成するものをマルチパスコンパイラという。これに対し1回のパスで、ソースプログラムからオブジェクトプログ

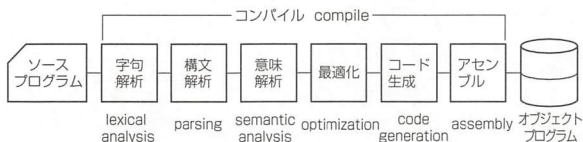
ラムに変換できるものをワンパスコンパイラという。

(4) 拡張した言語仕様で記述したソースプログラムを標準仕様のソースプログラムに変換するプログラムをプリコンパイラ、またはプリプロセッサという。Cでは、プリプロセッサとコンパイラの2段階から構成されている。

(5) 通常のコンパイラよりコンパイル工程が複雑で、コンパイル時間はかかっても、実行時間の短縮を重視する機械語コードを生成するコンパイラを特に最適化コンパイラという。

(6) エラー検出、自動修正の機能を強化したコンパイラを特にチェッキングコンパイラという。

(類)トランスレータ、プリプロセッサ

コンパイル[compile]

プログラム言語で記述したソースプログラムを、コンピュータが実行できる機械語コード(オブジェクトプログラム)に変換する処理。翻訳ともいう。この処理を行うプログラムをコンパイラという。高水準言語で記述したソースプログラムと同等の処理を行う機械語プログラムを作り出す過程は複雑で、数段階に分けて中間的なコードを作って行うのが一般的である。(参)翻訳プログラム、字句解析、構文解析、意味解析

コンピュータウィルス[computer virus]

自己増殖能力と潜伏能力、感染能力をもち、ネットワークや記憶媒体を経由して侵入し、システムに悪影響を与える悪性のプログラム。単にウィルスともいう。発見が困難で、プログラム利用のたびに伝染性をもつことから、ウィルスという名を比喩的につけた。ウィルスによる被害をやはり比喩的に感染、発病などという。ウィルスを発見し、無効にするアンチウィルスプログラムを、やはり比喩的にワクチンという。

ウィルスには、プログラムファイルに感染するもの、ハードディスクの

ブートセクタに感染するもの、マクロプログラムの文書ファイルに感染するものなどがあり、ファイルの破壊、データの改ざんや盗用、仕様以外のメッセージ出力や画面の破壊、ネットワーク上に異常データの大量送出などを行う。

ネットワークシステムが複雑になると感染経路の特定も難しくなる。比較的有効な対策は、最新バージョンのワクチンプログラムを定期的に実行することである。コンピュータネットワークで電子メールを介して感染するウィルスも現れており、油断ができない。出所が不明なプログラムは実行せず、文書は開かないで捨てるのが賢明である。

コンピュータシステム[computer system]

狭義には、中央処理装置、主記憶装置、制御装置、入力装置、出力装置、通信装置などをもつ処理装置の集まり。広義にはソフトウェア、さらにコンピュータ利用の目的で定めた規則や組織化した人材の集団まで含む。

コンポーネント[component]

構成要素のこと。VSAM編成ファイルでは、レコードを記憶するための連続した複数のCA(制御域)をコンポーネントと呼ぶ。ESDS、RRDSでは、クラスタはデータコンポーネントだけから構成され、KSDSのクラスタは、データコンポーネントとインデックス(索引)コンポーネントから構成される。

サービスプログラム[service program]

システム運用やユーザプログラムの実行を助けるプログラム。システムの制御や結果の生成に直接関係はもたない。ユーティリティプログラム、デバッギングエイドなどを含む。(関)オペレーティングシステム

最下位桁[least significant bit：LSB] (同)LSB

再帰[recursive／recursion]

問題を分割統治の手法で分割したとき、分割した部分に、同じ分割統治の手法を繰返し適用すること。アルゴリズム(関数、または手続き)が再帰的であるとは、その内容を定義する際に、自分自身をその定義中で使用することをいう。例えば、サブルーチンの実行中に自分自身をサブルーチンとして呼び出すこと。関数が自分自身をコールすること。

Fortran、COBOLでは使えないが、PL／I、Pascal、Cでは使える。サブルーチンの再帰的な呼出しは数式の構文解析や順列の生成などで有効な技法である。

最上位桁[most significant bit：MSB] (同)LSB

最上位モジュール[main module]

階層構造の最上位のモジュール。具体的には、プログラムの起動により最初に呼び出されるモジュール、プログラムの入口となるモジュールである。例えば、カウンタの初期化、ファイルのオープンとクローズ、各機能(モジュール)の制御などの処理は、一般に最上位モジュールで行う。これに対し、呼び出されるモジュールを下位モジュール、従属モジュールという。

(類)メインプログラム、(対)従属モジュール、(参)セグメント

再使用可能プログラム[reusable program]

一度主記憶上にロードされた後は繰り返し実行できるルーチンまたはモジュール。その実行中に自らの内容を変更しないように、あるいは実行中に変更した場合は初期状態にもどるように作ってあり、再ロードしないでも使用できる。実行中に自分自身を修正、変更して終了するプログラムは、再使用できず、再び原典からロードしなおさなければならない。これらは再使用不可能(不能)プログラム(non-reusable program)という。

あるタスクがそのルーチンまたはモジュールを使い終わって初期化がすめば、別のタスクが再使用できるプログラムを逐次再使用可能プログラムという。逐次再使用可能プログラムでは初期状態にもどるまでは、他のタスクに使用させないように排他制御を行う必要がある。

また実行中に自らの内容を変更せず、ひとつの制御の流れ(通常タスク

という）が抜け出ないうちに次の制御の流れが入ることを許すプログラムを、特に再入可能プログラムという。（参）再入可能プログラム

最適化[optimization]

①ある系について一定の状況のもとで可能なあり方の中から、最も好ましいものを選び出すこと。あるいは、最も好ましい状態になるように系のパラメタを調整すること。

②ハードディスク上のデータの記録を整理、再配置して記憶可能な空き領域を増やすディスクの最適化は、デフラグメントという。

最適化コンパイラ[optimizer／optimization compiler]

処理スピードが速く、コンパクトで能率のよいオブジェクトプログラムを生成するコンパイラ。与えられた原始プログラムを解析して、処理の重複や無駄を発見・修正する。最適化が行われると、データが暗示的に書き換えられたり、変数を選択的にレジスタに割当てたりということが起こる。

実行時間の短縮と記憶領域の縮小は必ずしも両立しないが、一般には実行時間の短縮を優先する。現在、主流のコンパイラはオプション機能として最適化コンパイルを具備しているものが多い。最適化にはプログラムの全般にまたがるグローバル（大域的）な手法と、計算式の変形などのローカル（局所的）な手法とがある。最適化は生成したプログラムの実行時間を短縮することであり、最適化コンパイラのコンパイル時間は普通のコンパイラよりも増加する。

サイトチェック[sight check]

入力したソースデータの正確さを調べるために、ダンプ内容を目で見て調べるデータチェック方式。目視検査ともいう。

サイトライセンス[site license／site licence]

ひとつのソフトウェアを企業内、学校内などの範囲で複数のユーザが利用する権利。または、研究や教育の場に限り複数の不特定利用者に割安の料金で使用許諾すること。パッケージソフトの販売契約、課金方法の一形態。同一ソフトウェアを複数のコンピュータで利用する場合、端末数や最大同時利用数に制限されず、決められた範囲内（部署、事業所など）で無制限にインストールできる。

再入可能プログラム[reentrable program]

複数のプログラムが同時に使用できるような構造をしていて、繰り返し実行できるルーチンまたはモジュール。その実行中に自らの内容を変更しないため、先行のタスクが終了する前に次のタスクが利用できる。このようなルーチンやモジュールを再入可能プログラムという。同時に同一のプログラムを共用することにより、記憶領域の節約や、処理効率の向上を図ることができる。実時間処理で多く用いる。（参）再使用可能プログラム

再配置可能プログラム[relocatable program]

主記憶装置内のどの領域にロードしても実行可能なプログラム。相対アドレスをもっていて、主記憶に格納する際に絶対アドレスに変換する。プログラムを主記憶装置内の任意の位置に格納できるため、主記憶空間の有効な使用が可能になる。また、主記憶空間を複数の領域に分け、複数のプログラムを記憶させる場合、そのプログラムは再配置可能プログラムであることが望ましい。

再編成[reorganization]

あらかじめ確保したファイルの領域が効率的に使えないときに、ファイルを作り直す処理。索引順次編成ファイルやネットワーク型データベースなどで必要となる。再編成が必要なのは次のような場合である。

(1) 領域の無駄使いにより、新しいレコードを追加できなくなった場合
(2) 使用できない断片的な記憶空間が増加した場合
(3) 一連のレコードが非連続的な領域に分布しアクセス効率が低下した場合

(参)索引順次編成、ガーベイジコレクション、(類)データベース再構成

再利用技術[reuse technology]

ソフトウェアの再利用のための技術。ソフトウェア生産の効率を高める技術として注目される。大きく次の二つの技術に分けられる。

(1) ソフトウェアそのものを再利用できるモジュールの形にしておき、新たなソフトウェアに組込む。ソフトウェアの開発量を抑制できる(狭義の再利用技術)。
(2) ソフトウェアを作る過程を標準化し、自動化できるツールを開発しておく。ソフトウェアの開発時間を縮小できる(自動化技術)。

先入れ先出し[first in first out] (同)FIFO

作業領域[work area] (類)ワークエリア

索引[index] (同)インデックス

索引域[index area] (関)索引順次編成

索引順次編成[indexed sequential organization]

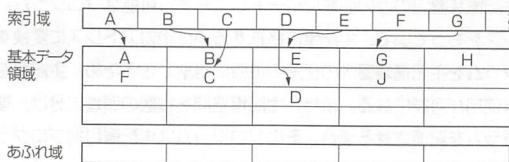
乱処理と順次処理の両方を能率よく行えるファイル編成の一方式。ファイル中のレコード内のある項目をキーとし、キーの指定によってアドレスを決め、レコードを読み書きする。単に索引編成ともいう。すなわち、レコードを格納した相対トラック番号とそのトラック内のキーの最大値とをインデックステーブル(索引)で対応づけておく。キーを指定するとそのインデックステーブルを参照し、レコードの呼出しを行う。索引順次編成ファイルを直接アクセス装置上に作るときは、その記憶領域を3つに分ける。

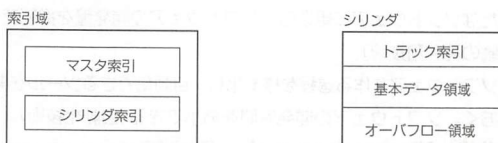
(1)基本データ領域(プライム領域):プライム領域はデータレコードとトラック索引を含む。この領域はすべての索引編成ファイルに存在す

る。ファイルを作成するときは、レコードをプライム領域に書き出す。

索引域　A　B　C　D　E　F　G

基本データ領域　A F　B C　E D　G J　H

あふれ域

（2）**索引域（インデックス領域）**：実際のレコードの補助記憶装置上のアドレスとレコード中の項目との対応を示すインデックス（索引）を記録する領域。索引域に作られるのはこのうち、シリンダ索引とマスタ索引である。

- **シリンダ索引**：各シリンダ内のレコードの最大キーとシリンダ番号が記録された索引。ひとつのファイルにひとつあり、検索対象のレコードがどのシリンダに存在するかを特定するために用いられる。
- **トラック索引**：シリンダごとに作られる索引。シリンダの各トラックにあるレコードの最大のキーの値とそのトラック番号が記録される。
- **マスタ索引**：シリンダ索引の索引。シリンダ索引が大きくなり、検索時間を要する場合に作成され、シリンダ索引のトラックの最大キーとトラック番号が記録される。

（3）**あふれ域（オーバフロー領域）**：レコードを追加するために、あらかじめ余分に確保した領域。レコードの追加があっても全ファイルを書き直す必要はなくなる。あふれ域には、各シリンダごとのシリンダあふれ域と、どの場所からでも利用できる独立あふれ域とがある。あふれ域に多数のデータが入った場合は能率が下がるので、ファイル本体を作り直したほうがよい。

- **シリンダあふれ域**：索引順次編成ファイルのオーバフロー領域（あふれ域）のうち、各シリンダごとのシリンダあふれ域。つまり、そのシリンダ内のトラックからあふれたレコードを収容する。なお、特にシリンダあふれ域を設けず、独立あふれ域のみを設けて、あふれたすべてのレコードをそこに収容する場合もある。
- **独立あふれ域**：各シリンダのシリンダあふれ域からもはみ出したレコードを収容するあふれ域。

(関)再編成

索引ファイル[indexed file/indexed organization file]

磁気ディスク装置やフロッピーディスク上に作成できるファイル。キー項目の昇順に呼び出す順呼出しと、乱呼出しの双方が可能である。キー項目についてレコードを記憶している場所を示す索引が設定され、呼出しを行うときにこれを参照する。なお、順ファイルに比べると大きな記憶領域を必要とする。

削除[delete/deletion/erase]

①広義には媒体内のファイルやファイル内のレコードを使用できなくする操作。誤って削除したデータを再度使用可能にすることを復活(または回復)という。OSやアプリケーションソフトによって、二つの場合がある。

(1)物理的(電子的)にデータ領域を無意味な文字(空データ)で書き換える場合、特に消去、抹消(erase)ともいう。消去した領域には、別のデータを記憶させることができる。EP-ROMでは強い紫外線にさらすことで記憶を消去する。磁気記憶媒体では、スペースコードに置き換える場合と、バルクイレーサという機械で完全に磁気記録を無くしてしまう方法とがある。

(2)データ領域が空き領域になったという情報を別に書き込んで削除したことにする場合があり、この場合は、データ領域に別のデータがオーバライトされるまでは、アンドゥコマンドやアンデリートコマンドなどで復活できることがある。

②文書処理では、指定した行や文字列を消す操作。空白文字で埋める場合と後ろの文字列で詰める場合がある。(対)挿入

サブシステム[subsystem]

大きな統一的なシステムの一部分の機能を分担するシステム。大きく複雑な機械でも、いくつかの機能をもつ部分から構成されるように、コンピュータシステムもより小さな複数のシステムから構成することができる。その小さなシステムもさらに小さなシステムの集合ととらえることができる。このような階層的な見方で、上位のシステムから下位のシステムを見るとき、サブシステムという。

サブスキーマ[subschema]

CODASYL型データベースを応用プログラムから見たときの、処理すべきデータおよびその論理構造。応用プログラムからデータ操作の呼出し定義時に指定(記述)する。サブスキーマ言語により、使用可能なデータベースの部分の論理的な構造、レコードの使用可能な構成要素、親子関係の使用可能な型を定義する。サブスキーマが応用プログラムから扱えるデータを制限するため、データの機密保護が確保できる。(関)スキーマ、データベース

サブディレクトリ[subdirectory]

MS-DOSやUNIX系OSなどの階層構造のファイル管理システムで、あるディレクトリのすぐ下位にあるすべてのディレクトリ。分岐ディレクトリともいう。サブディレクトリは、さらにその下位に複数のサブディレクトリをもつことができる。これらは独立のディレクトリ名をもつことができ、数多くのファイルのグループ化に利用する。

サ

サブメニュー[submenu]

階層構造をもつメニュー選択法のひとつ。あるメニューのすぐ下位にあるメニュー。

サブルーチン[subroutine]

①プログラムの論理的にまとまった機能部分で、必要な箇所で呼び出して使用するルーチン。処理の流れの中で共通機能を多数回利用する場合、その機能をサブルーチンにすることが望ましい。プログラムの記憶域、作成時間、労力が節約できるため頻繁に利用する手法である。

②プログラムの中でサブルーチンコールで呼び出されるプログラム部分。

サブルーチンはその性格から、(1)開いたサブルーチンと(2)閉じたサブルーチンの2通りに大別できる。

(関)開いたサブルーチン、閉じたサブルーチン

サブルーチンコール[subroutine call]

プログラムからサブルーチンを呼び出すこと、またはそのための命令や文。サブルーチンは複数のプログラムから何回でも呼び出すことができるので、システムの機能分割に活用できる。また、プログラム言語に依存せず、他のプログラムから呼び出して利用できるものもある。

サブルーチン副プログラム[subroutine subprogram]

Fortranの用語。SUBROUTINEステートメントから始まる外部サブルーチン。CALL文によって呼び出し、呼び出したプログラムの引数に値を返す。

サポート[support/user support]

①維持すること。コンピュータシステムなどにおいて、高度な専門知識を必要とするハードウェアやソフトウェアを維持、保守する体制。

(1)企業システムでは、開発部門、開発会社がユーザ部門の教育、指導、保守、リカバリなどに当たるケースがある。

(2)パソコンシステムでは、ハード・ソフトメーカの専任のアドバイザがユーザへの助言やアフタサービスに当たるケースがある。サポートはその内容によって無料と有料の場合がある。その窓口をサポートセンタという。

②あるソフトウェア製品が別のソフトウェアと矛盾なく共存できたり、機能を提供できること。例えば、アドインソフト、アドオンソフトや周辺

機器の機能拡張ソフト[A]が、既存ソフト[B]の機能を拡張するとき「AはBをサポートする」などという。

算術演算[arithmetic operation]

加算、減算、乗算、除算、べき乗などの演算。データの型によって、固定小数点演算、浮動小数点演算、10進演算などがある。論理演算に対比する語。(関)算術演算子

算術演算子[arithmetic operator]

プログラム中や表計算ソフトのセルの計算式(算術式)を書くとき、その式の中で使うことができる演算子。式の中に算術演算子が複数ある場合は、演算子間の優先順位と結合規則に従って演算する。

(1) かっこが多重にある場合は、内側のかっこ内から計算する

(2) かっこがない場合は、次の順序で優先的に計算する。

ただし、同じ順位の演算子が並ぶときは左側から順に計算する。

- 単項演算子の+、または-(正負の符号反転)
- べき乗(**や^)
- *、または/(乗除算)
- +、または-(加減算)

算術演算子	意味	例
+(プラス)	加算	A+B、X+4
-(マイナス)	減算	A-B、A-1
	符号反転	-A(Aの符号を反転する)
/(斜線)	除算	A/B、X/6
*(星印)	乗算	A*B、X*2
**(星印2個)	べき乗	A**3 X**Y

算術シフト[arithmetic shift] (参)シフト、演算命令

参照[reference]

①プログラムで、処理対象の名前を使用すること。引用ともいう。例えば、変数名や定数名を式の中で指定すること、配列要素の位置を指標や添字で指定すること、プログラム中で関数名を使用すること。参照により、構成要素をパラメタで一意に指定したり、同一手続きが利用でき、プログラムを単純化できる。

参照方法には、変数名や定数名などの名前指定、配列要素の位置を指標や添字による指定、データを代表するコードによる指定などがある。

②データベースやファイル中の指定したデータを得ること。アクセスと同義。

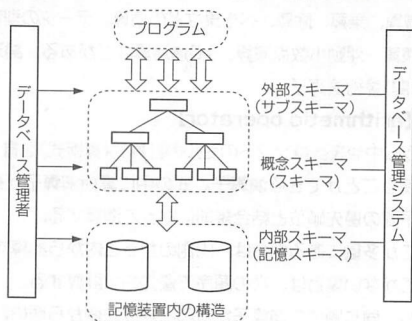
3層スキーマ[three schema structure]

ANSI/X3/SPARCが提唱した3層のスキーマ。

(1) 概念スキーマ:コンピュータや利用分野などのことから離れて、事実の表現としてのデータの意味や関係の構造を記述するもの。

(2) 外部スキーマ:データベース内のデータを操作する応用プログラムご

とに、データの格納構造を記述するもの。

(3) **内部スキーマ**：概念スキーマと外部スキーマを、コンピュータや記憶装置上で実現するための物理的な構造を記述するもの。

三文字表記[trigraph sequence]

Cの用語。キーボードに対応する記号キーがなく、入力ができない場合に、対応する記号を表現するもの。記号3文字で1文字を表現することからこの名がある。三文字表記はすべて'??'と1文字によって記号を表す。

シ

シーケンシャルアクセス[sequential access] (同)順呼出し法

シーケンシャル処理[sequential processing] (同)順次処理

シーケンシャルファイル[sequential file] (同)順編成ファイル

シーケンスコード[sequence code] (同)順番コード

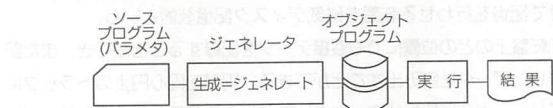
シーケンスチェック[sequence check] (同)順番検査

シームレス化[aim of seamless]

企業内システムで、部門間でのデータの受け渡しや処理を連続的にスムーズに行うこと。一般に、部門が異なるとき生じやすいデータの再入力や処理の分断を抑制すること。一連の業務が多部門にわたるとき、データ処理の流れの中で部門ごとにデータを転記したり、再入力したりする手間を要さないように、電子化データをコンピュータ間で受け渡し、流れを分断させないこと、または、データ処理の流れを連続的に行うこと。

ジェネレータ[generator]

①あらかじめプログラムの骨組みができており、利用者が入力データや処理結果の内容と形式、および処理条件などを一定の書式の各欄に記入、

入力すると、自動的に処理に必要なプログラムを作成してくれるプログラム。生成プログラムともいう。(例)RPG

②データを作成するユーティリティプログラム。テストデータジェネレータなど。

シェルソート[Shell's sort/diminishing increment sort]

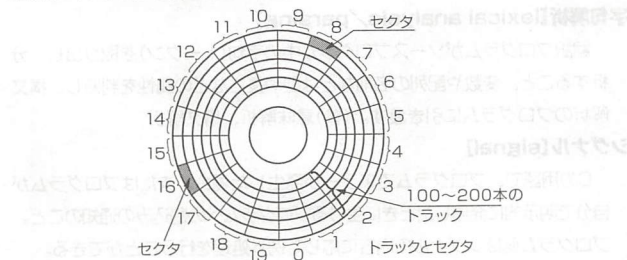
内部ソートのアルゴリズムのひとつ。基本的な考えは、挿入ソートが大半のデータが正順で一部データが乱順である場合に効率がよい性質を利用し、まず大ざっぱに整列し、最後に挿入ソートで仕上げる方式である。

配列中のデータから、まず一定間隔(例えばm)ごとにデータをとって整列する。これをずらしながらm回繰り返せば、全データをmグループに分け整列したことになり、大まかには整列される。間隔幅を変えて大まかな整列を繰り返せば、整列の状態に近づく。そこで挿入ソートを行えば、効率よくソートが完了するというアルゴリズムである。実験的には、データ量の1.25乗に比例する平均計算量が必要である。

時間計算量[time complexity]

アルゴリズムの効率を評価する目安のひとつで、処理の終了までに要する実行時間の長さ。一般的にはハードウェアの速度に依存しない議論をするために絶対的な時間ではなく、実行ステップの回数やデータがn倍に増えた場合のステップ数の増加の割合で議論する。同じアルゴリズムであっても、入力した値によって手間が異なる。そのアルゴリズムにとって最も時間を要する値を与えたときの計算量を最大時間計算量(worst case time complexity)といい、考えられるすべての入力の計算量の平均を平均時間計算量(average time complexity)といい、計算量を示す目安とする。

磁気ディスク[magnetic disk]

シ

磁性体を両面に塗った回転円盤で、その表面を磁化しデータを記憶する媒体。高速回転する磁気ディスクに、アクセスアーム、磁気ヘッドと組み合わせて記憶を行わせる装置を磁気ディスク記憶装置という。

(1) 回転盤上のどの位置にでも直接データを記録することができ、また記録したデータを読み出すことができる。円盤を同心円上のトラックに区分し、さらに放射状のセクタに区分し、それぞれにアドレスをつける。このため、磁気テープに比べ多様な処理が可能になる。

(2) 円盤の素材が金属、ガラスなど剛体であるものをハードディスク、合成樹脂の柔らかい種類をフロッピーディスク、またはフレキシブルディスクという。

磁気テープ[magnetic tape：MT]

プラスチックをベースとしたテープに磁性材料を塗布した記憶媒体。磁化の方向によってデータを記憶する。長いテープを巻いて使うため順呼出しのみが可能である。主としてバックアップ専用の記憶媒体として用いる。表面の欠陥、異物の付着で信号の誤り（ドロップアウト）が生じる欠点がある。ケースに収納したまま扱えるカートリッジテープが主流であり、主に、大量のデータ保存（ストリーマ）として利用する。

識別子[identifier：ID]

広義には、文字や数字を、一意に識別できるように組み合わせた文字列。

①プログラム中でのファイル名、変数名、定数名、装置名、段落名などの名前。それぞれの範囲内で同一の文字列であってはならない。

②データ通信では、ある機能、エンティティ、ユーザ、ホストコンピュータ、自端末と相手端末、装置タイプ、レベルなどを識別するために、それぞれ分類ごとに決めたユニークな（一意の）番号または記号。ID（ホストID、端末ID、ユーザIDなど）と呼ぶ場合もある。

字句[lexical unit／lexical element]

プログラムを構成する意味をもつ言語要素。字句要素ともいう。例えば、予約語、変数、定数、区切り符号、演算子、行番号、識別子、注釈など。

字句解析[lexical analysis／parsing]

翻訳プログラムがソースプログラム中の字句（トークン）を取り出し、分析すること。変数や配列の名前などは型や長さなどの属性を判断し、構文解析のプログラムに引き渡す。（参）意味解析、構文解析

シグナル[signal]

Cの用語で、プログラムでエラーが発生した場合、またはプログラムが自分で明示的に発信したときに生じるプログラムへの割込みの通知のこと。プログラム側はシグナルの内容に応じて例外処理を行うことができる。

(参)割込み

資源[resource] (同)システム資源

資源管理プログラム[resource management program]

コンピュータのシステム資源を効率よく管理するためのプログラム。限られた資源を効率よく使用し、スループットの向上やコンピュータシステム利用コストの低減を図る。コンピュータシステムで、CPU、メモリや磁気ディスク記憶装置等の各種入出力装置をハードウェア資源、プログラム等をソフトウェア資源という。オペレーティングシステムの役割のひとつにこれら資源の割当てや管理がある。

自己解凍形式[self-extracting form]

実行可能なプログラムを圧縮したファイル形式のひとつ。実行コマンドを受けると自動的にファイルを展開し、実行するプログラムを付加したもの。解凍用のプログラムを利用する必要がなく、パソコン通信などでプログラムを授受するのに便利である。

事後保守[corrective maintenance/breakdown maintenance]

故障が起こった後でシステムを運用可能状態まで回復する作業。故障の時間的分布が均一のアイテムは、予防保全が不可能なので事後保守にならざるを得ない。この場合、緊急性が高いので、緊急保全ともいう。

四捨五入[round off]

数の丸め方の一手法。削除する数字の最上位桁がその数字位置(桁)の基数の半数の半分以上の値であるときに限り、残った数表示の部分を、その最下位数に1を加えて必要な桁あげを行うこと。

COBOLでは、COMPUTE C ROUNDED=A/Bというように、受取り側データ項目名の直後に四捨五入指定ROUNDEDを使うことにより四捨五入できる。またFortranではNINT、ANINT関数を使うことにより四捨五入できる。表計算ソフトでは、表示形式の指定で行う。

(参)整数化、(類)丸め

システムアドミニストレータ[system administrator]

情報システムの利用者の立場で、エンドユーザコンピューティング(EUC)の推進に従事し、EUC環境の充実とツールの利用を推進する技術者。利用者側において情報処理に関する一定の知識、技術をもち、部門内あるいはグループ内の情報化を利用者としての立場から推進する。具体的には、以下のような役割をになう。

(1)エンドユーザコンピューティングの推進

(2)部門内あるいはグループ内の情報システムの構築またはその支援

(3)情報システムの提供者側に対する利用者側の意見要望の提起

(4)情報システムの運用とシステム利用環境の整備

システム運用[system operation]

ユーザによるシステムの稼働、システム資源の管理、システムの機能、性能を継続的に維持する作業。広義には運用設計、運用基準の策定、サービスの提供を含む。一般的な運用管理は、次のような領域で考える。

(1)**性能管理**：規定の処理能力の維持・性能の測定や評価
(2)**障害管理**：障害の監視・検知・原因や影響の把握・回復や保守
(3)**運用管理**：運用の合理化や自動化・運用経費の算出や課金
(4)**機密管理**：システムの情報資源の悪用や盗用の防止

システムエンジニア[systems engineer：SE]

システムを構築するとき、要求定義、現行システムの調査分析、新システムの概要設計、詳細設計までの職務を担当する技術者の総称。要求定義の作成に従事する人をシステムアナリスト、システム設計に従事する人をシステムデザイナと分ける場合もある。通産省の指導で作成した高度情報化人材育成標準カリキュラムは、さらに専門分野によって細分化している。

システムオペレータ[system operator]

①広義には、コンピュータシステムの操作員。コンピュータを起動したり、プログラムを実行したり、媒体を補助記憶装置にセットしたり、計算結果を取り出すなどの作業を行う。単にオペレータともいう。
②パソコン通信のBBSや電子会議をホスト側で運用、管理する人間。オンライン会議やフォーラムの世話人。略してシスオペ、またはSIGオペともいう。シスオペは、参加者の募集、登録、参加者の質問に対する回答、ボードの管理、電子会議の議長、相互のコミュニケーション円滑化への誘導、コンピュータの資源管理などを行う。

システム開発[system development]

ユーザの要求を把握し、システム設計、プログラム設計、プログラム開発、システム運用までの工程を管理し、実現すること。大規模なプログラム開発と同義に用いる。総合的、広域的にわたる開発をさすことが多い。
(類)プログラム開発、(類)システム設計

システム構成[system configuration]

コンピュータシステムでは、中央処理装置、主記憶装置、外部記憶装置、通信制御装置などの組合せを意味する。
(例)シンプレックス構成システム、デュアル構成システム、デュプレックス構成システム、タンデム結合システム

システムコール[system call]

①利用者やシステム内のプログラムが、オペレーティングシステムが提供

するサービス機能を呼出すこと。利用者はファイルアクセス制御やプログラム管理などの処理のために、自分で複雑なプログラムを作成しなくてもシステムコールを利用して行うことができる。(参)SVC

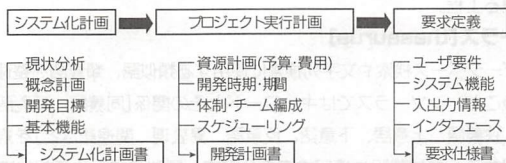
②UNIXやOS/2の応用プログラムがOSの機能を呼び出す関数。他のOSでは、システムサブルーチンやAPI(application program interface)などと呼ぶ。入出力制御、システム管理、データ管理、データ通信などの機能を呼び出せる。

システム資源[resource]

コンピュータで処理を行う際に使用するハードウェア、ソフトウェアの時間や容量、労力、エネルギー、資材等の総称。オペレーティングシステムがこれらの資源を管理し、使用可能な主記憶領域や出力装置、磁気ディスク記憶装置の残り容量等をオペレータに示す。マルチプログラミングシステムでは、これらの資源を最も有効に利用すべくジョブ管理を行っている。

システム設計[systems design]

コンピュータシステムの目的、機能、性能を設定し、実現するための具体的方法を設計すること。ハードウェア、ソフトウェア、時間、要員等の資源をどう配分し、構築し、運用するか、方針を決定する。システム設計の手順は開発するシステムの特性や規模によって異なる。

基本的には、新システムの目標の設定、外部設計、内部設計などプログラム開発に先立つ上流工程の基本計画、設計作業をさす。

システムソフトウェア[system software]

コンピュータシステムの一般の利用者のために、あらかじめ用意してあるプログラムの総称。システムソフトウェアは、その役割から次の二種類に分類できる。

(1) **基本ソフトウェア**:広義のオペレーティングシステムとほぼ同義。入出力や実行制御、資源を確保する制御プログラム、コンパイラなどの言語プロセッサ、プログラム支援ツールや各種標準ソフトウェアパッケージなどのユーティリティプログラムなど

(2) **ミドルウェア**:応用ソフトに基本的機能を提供するソフトウェア。通信管理、GUI制御、グラフィック処理プログラムなど

(対)アプリケーションソフト

システムダウン[system down]

ソフトウェアの異常やハードウェアの故障により、システムの機能が予期せず停止すること。システムダウンすると、システムのサービスが完全に停止したり、完全に停止しなくてもサービスの品質が極端に悪くなる。大規模オンラインシステムのシステムダウンは社会に大きな影響を及ぼす。そのため、デュアルシステムやデュプレックス構成システムなどに比べ、さらにシステムダウンの発生率を大きく低下させたフォルトトレラントシステムに関心が集まっている。

システムテスト[system test]

開発したシステムが、総合的に見てシステム設計段階の目的、目標を実現しているか確認するテスト。総合テスト(integrated test)ともいう。おもなテスト項目は、機能テスト、過負荷テスト、安定性テスト、耐障害テスト、性能テストなどであり、運用の現場にシステムが移行された場合を想定して開発者が行うテストである。オンラインシステム、バッチシステムの両方で構成されたシステムでは、資源の競合が相互に影響を及ぼすので、実際の運用に即したテストが必要である。(参)受入れテスト、結合テスト

システム分析[system analysis]

情報システムの構築に先だって、経営課題、事業環境、業務改善策などを考慮しながら、システム化計画を検討すること。現状分析、要求分析、情報システムの定義、日程、コストなどのデータから、システム開発計画を練る工程。

シソーラス[thesaurus]

データベース検索や文字列検索に使用する類似語、類義語を整理した辞書のこと。シソーラスではキーワードとその関係[同義語、類義語、広義語、狭義語、上意語、下意語、反意語、異表現、関連語など]を規定して配列する。機械翻訳や情報検索の分野で、語群の統一や情報検索のキーワードの融通性として重要な概念である。コンピュータで自然言語による情報検索をおこなう際にユーザの要求に該当する必要十分な情報を得るためには、シソーラスを参照したキーワードを使って効率のよい論理演算式を作る必要がある。

実アドレス[physical address／real address]

主記憶装置上のアドレス。論理アドレス(仮想アドレス)に対比して使う語。物理アドレスともいう。実アドレスで表す記憶装置を実記憶とよび、論理アドレスで表す記憶装置を仮想記憶という。

大きなサイズのプログラムや複数のプログラムを実行するには、大きなアドレス空間を制御するCPUとそれに対応する大容量のメモリが必要とな

る。この要求を経済面から解決する方法として仮想記憶方式がある。プログラムは主記憶の容量やアドレスに無関係な大きな仮想記憶内で作り、実行時にはじめて主記憶に移し、実際の演算を行う。仮想記憶上の論理アドレスを参照すると、動的アドレス変換機構がアドレス変換テーブルにより実記憶装置上の実アドレスに変換する。

（関）仮想記憶、（類）実効アドレス

実記憶装置［real storage］

仮想記憶システムにおける主記憶装置のこと。物理的には、実記憶装置と主記憶装置とは同一である。しかし概念的には、実記憶装置は仮想記憶システムの利用者にとって利用可能なアドレス領域の部分だけを意味する。

（参）仮想記憶

実行［execution］

①コンピュータシステムによって一連のコマンド、またはプログラム中の複数の命令や文を自動的に遂行する処理。広く、コマンドを解釈して処理することから、プログラムによる処理結果を得ることまで含む。

②GUI環境では、指定した処理を行うコマンド、またはアイコン。文字列の一括変換・削除など処理の対象に重大な影響を与える操作、多くのパラメタを指定した後に初めて実行できる処理においては、ユーザの操作ミスを警戒して指定を即時に実行せず、再確認を求める場合が多い。実行コマンド・実行ボタンの入力があってから処理を行う。

実効アドレス［effective address］

インデックスアドレス指定や間接アドレス指定などのアドレス指定を行ったあとのアドレスのこと。有効アドレスともいう。主記憶装置の実アドレスとして、そのまま実際の記憶場所に対応して使用する場合と、仮想記憶のように実アドレスを論理アドレスとして、さらに補助記憶装置を含めたアドレス空間に対応させる場合がある。

実行可能プログラム［executable program／execute file］

コンパイル、リンクを行い、コンピュータを動作させる準備がすべて完了したプログラム。ロードモジュールはこれに相当する。

実行サイクル［execution cycle］

逐次制御計算機で、命令デコーダが命令を解読してから、処理に必要な制御信号を発し、処理指示を記憶し実行を完了するまでの段階。命令実行段階ともいう。（参）命令サイクル

実行文［executable statement］

プログラムの主要な構成要素で、データに一定の処理を施したり、実行順序を制御する文。前者には、代入文、入出力文などがあり、後者には、繰返し文、条件文などがある。（対）非実行文

シ

実時間処理[real time processing]　(同)リアルタイム処理

10進分類コード[decimal classification code]

10進数を用いるグループ分類コード。ある集合の要素を10個以下のグループに分類し、さらに各グループ内の集合の要素を10個以下のグループに分類し、各桁に数字を与えることを繰り返すコード化技法。代表的な例には図書分類のNDCがある。下位の桁を増やすことで細分化が可能である。

(例)NDCではコンピュータ文献を549.92に分類する。

500　工学
540　電気工学
549　電子工学
549.9　電子装置の応用
549.92　電子計算機

実引数[argument]　(関)引数

シノニム[synonym]

直接編成ファイルでファイル中の欠番キーが多く、直接アドレスによる方法が不向きな場合、アドレス生成法(除算法、数字分析法、重ね合わせ法、基数変換法等)のハッシュ関数を用いて実際に格納するアドレスの範囲を圧縮する。このとき複数のレコードの異なったキーを同一のアドレスに変換してしまうことがあり、これをシノニムという。

キー項目を変換して得たアドレス(本来のアドレス)をホームアドレスといい、ここに入るレコードをホームレコードという。ホームアドレスをすでに他のレコードが占めているとき、シノニムレコード(あふれたレコード)を格納するアドレスをシノニムアドレスという。

(参)ハッシング、直接アドレス方式

指標[index]　(同)インデックス、索引、添字

シフト[shift]　(参)シフト演算命令

シフトJISコード[shift JIS code]

パソコンなど小型コンピュータのOSで使用する漢字コードの一体系。本来のJIS漢字コード体系の要求する漢字シフトコードを使用しないですむよう改良した業界標準の漢字コード。JISコードでは、1バイト系の文字(半角文字)と2バイト系の文字(全角文字)が混在している場合、それらの区別をつけるためにエスケープコードのKIとKOが必要である。

シフトJISコードは漢字1バイトめを1バイトコードと重複しない空きコード(未定義部分)を使用しているため、ある1バイトを見れば漢字か漢字でないかの区別がつく。これによりKI、KOコードを不要とした。ただし、シフトJISコードは日本のパソコンのローカルなコード体系であり、国際標準のコードではない。空きコードを別の意味(周辺機器の制御コードな

ど)に使っている装置とはそのままでは共存できないなどの問題がある。

シフト演算命令[shift instruction]

汎用レジスタ(GR)において、語の中のビットを指定した桁だけずらす命令。シフト方向で右シフト、左シフトとがあり、また、符号ビットの扱いの差で論理シフト、算術シフトに分かれる。この他、循環シフト命令などがある。

(1)算術シフト(arithmetic shift)

符号部分(MSB)を除いて行われるシフト演算。シフトにより空いた部分には、左シフトのときは0が、右シフトのときにはMSBと同じ値がセットされる。1ビット算術左シフトすると、GRの数値が2倍になる。ただ、符号ビットと、シフトで外に出るビットの2進数字が同じでないと、オーバフローにより、2倍にならない。逆に1ビット算術右シフトすると、GRの数値は2分の1になる。オーバフローは起きないが、右端から外に出たビットは無視され、切り捨てられる。

(2)論理シフト(logical shift)

符号部分を含めて行うシフト演算。シフトにより空いた部分には、左、右シフトとも0がセットされる。シフトの結果によりフラグレジスタ(FR)を設定する。

論理シフト命令は、乗除算の計算にも使えるが、左シフトの時のオーバフロー、右シフトの時の切捨てに注意が必要。また任意のビットデータの抽出や条件分岐命令との併用で、特定のビットの1か0かの判定等にも使う。

(3)巡回(循環)シフト(cyclic shift)

上位桁と下位桁を入れ換えるシフト演算。シフトにより空いた部分にはみ出した部分のビットをセットする。

シミュレータ[simulator]

物理的な現象または抽象的なシステムのふるまいをモデル化し、模擬実験を行うためのハードウェアまたはソフトウェアのシステム。実機を用いないシミュレータの例には、自動車を破壊することなく衝突時の安全性を検証したり、宇宙飛行士や航空機パイロットの養成訓練に使用するトレーナなどがある。株価の変動のような経済予測にも用いるのもシミュレータである。

またパソコンではシミュレーションゲームをさし、飛行機の操縦を疑似体験するフライトシミュレータや、政策を決定しながら都市を構築したり、水槽に熱帯魚を飼育していくゲームなどがある。

ジャーナルログ[journal log]

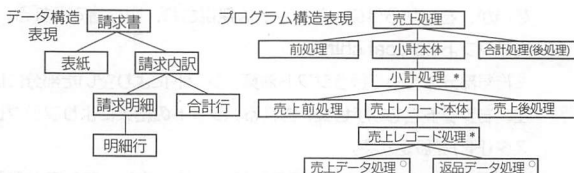
コンピュータシステムの動作、状態、操作状況などの時間的推移の記録。オンラインシステムなどで、どのような状態でシステムがダウンして

も回復できるように、処理途中のファイルやデータベースの更新情報を記録しておくファイル。システムの稼働統計をとるときなどにも利用する。定期的にとる稼働記録をジャーナル(journal)、ログファイル(log file)といい、これらの記録をとることをロギング(logging)という。トランザクション処理では、ジャーナルの採取は必須である。

シ

射影[projection]　(関)関係代数

ジャクソン法[Jackson structured programming]

1970年代半ばに Michael Jackson によって提案されたデータ構造と処理構造の図式的表現法。データ構造からプログラム構造(処理構造)が決まるとしている点で、データ本位の構造化手法である。このために4つの基本図式を用い、入力データと出力データの構造を書き、この関係からプログラム構造を書く。この際に用いるダイアグラムをジャクソン図という。

ジャンプ[jump／GOTO]　(同)飛越し

集合演算[set operation]

関係データベースの二つ以上の表に対する操作は、演算として定義できるが、この演算系の集まりをいう。この演算を適用することによって、複数の表から新しい関係を作ることができる。集合には次の3種類の基本演算がある。

(1)複数の表のすべての要素を集める和演算(union operation)

(2)複数の表の共通する要素を集める積演算(intersection operation)

(3)ある表の要素から別の表の要素を除いた要素を集める差演算(difference operation)

集合操作[set operation]

SQLのDMLで、ひとつの命令で複数のレコードをまとめた集合(set)を処理できる命令群(SELECT、INSERT、DELETEなど)がある。この命令による複数レコードの操作のこと。

修飾[modify／decoration]

作用する内容(データ)またはそのアドレスを修飾子により変化させること。変更ともいう。

(1)アドレス修飾：修飾子はレジスタやカウンタの内容であることが多

い。特にアドレスを修飾するときアドレス指定ともいう。

(例)インデックスアドレス指定(インデックス修飾)、ベースアドレス指定(ベースレジスタ修飾)、(関)アドレス修飾、アドレス指定

(2)型修飾:Cにおいては、型を型修飾子によって修飾すること。

修飾子[index] (同)インデックス

従属セグメント[slave segment]

(同)処理・加工セグメント、(対)制御セグメント

従属モジュール[slave module]

プログラムの構造化設計でモジュール分割したとき上位のモジュールから呼び出されるモジュール。下位モジュールともいう。

終値[terminal parameter/terminal value]

繰返しループを実行するに先だって、制御変数、領域に設定する定数、条件の値のひとつ。制御変数の値がこの値を越えると繰返し処理が終わる。(対)初期値

16進コード[double byte character code]

(類)区点コード、シフトJISコード

16進数[hexadecimal number]

16進法で用いる数字、または16進数を用いて表した数値。4ビットを1桁で表す。0~9の数字とA~Fの16種類の数字を用いる。

16進法[hexadecimal notation]

16を基数とする数の表記法。16進数の1桁は2進数の4桁(4ビット)に相当する。1バイト(2進数の8桁)を表すのに2桁ですむ。コンピュータの機械語やアドレスの表現に適し、メモリやファイルのダンプリストなどで使用する。(関)基数記数法

主記憶装置[main storage unit:MSU/main memory unit:MMU]

命令またはデータを、引き続く実行または処理のために記憶する装置。次のような特性をもつ。

(1)主記憶装置には記憶できる1語ごとにアドレス(絶対番地)をつけてあり、CPUは、メモリアドレスレジスタとメモリデータレジスタを介して、指定したアドレスの1語をロード、ストアする。つまり、プログラムによりアドレスで格納場所を指定できる。

(2)主記憶装置はアクセスタイムが短いほどよい。現在はICメモリ(MOS型)が中心で、数十ナノ秒程度である。

(3)記憶容量も大きいほうがよく、数百メガ~数十ギガバイトである。パソコンでは容量64~128メガバイト程度が一般的である。

(4)入力したデータは、必ず主記憶装置に記憶し、処理途中のデータも主記憶装置に保存する。電源を切ると記憶内容が消える性質があり、こ

シ

れを記憶の揮発性という。

(5) 主記憶装置は高速であるが比較的容量も小さく揮発性であるため、プログラムやデータの必要部分のみを主記憶装置に読み込み処理する。これを補うためにハードディスクなどの補助記憶装置がある。

(6) 主記憶装置はその役割から、内部記憶装置、実記憶装置と呼ばれることがある。

出力設計[output design]

コンピュータシステムが、いつ、だれに、どのような情報をどんな媒体で出力するのか設計する工程。データ処理の出力では、わかりやすく、必要十分な情報をタイミングよく提供する出力設計が重要になる。具体的な設計の範囲は、出力機器、媒体の選定、印字、表示のレイアウトから出力の配布、配送法(例：配布先の印字、袋綴じ印刷など)まで多岐にわたる。(対)帳票設計、(類)報告書設計、画面設計

出力装置[output unit]

コンピュータの基本装置のひとつで、出力機能を果たす装置。データの処理結果など、システムが保持するデータを外部に送り出す装置。一般には人間が認識できる形式で出力するプリンタ(印字装置)、ディスプレイ装置(表示装置)、音声応答装置などをさす。コンピュータ制御の工作機械、ロボット、通信の相手先なども出力装置に含める。

出力帳票[output／output sheet／report]

人間の読める形式に、データを印字したもの。その利用目的、印字装置の制約などを考え、用紙サイズ、レイアウト、プレプリントの有無、フォントなどをくふうする。(参)報告書設計

出力媒体[output data medium]

コンピュータの処理結果を出力装置により記録する媒体。人間が読みうる形式の媒体と、記憶媒体上に記録して別の出力装置により人間が読みうる形にする媒体がある。前者の例には、ディスプレイ装置、出力帳票が、後者の例には、磁気ディスク、フロッピーディスクなどがある。

主プログラム[main program]　(同)メインプログラム

巡回シフト[cyclic shift]　(参)シフト演算命令

巡回リスト／循環リスト[circular list]　(関)リスト構造

順次アクセス[sequential access]

あらかじめ定められた順序(例えば発生順、記録順)に従い、順番にアクセス、処理していく方法。磁気テープ上のファイルに対するアクセスなど、全レコードを順番に処理する場合に効率のよいアクセス方法である。ただし、特定のレコードにアクセスする場合もその前に位置するレコードをすべて読む必要がある。(対)直接アクセス

順次アクセスファイル[sequential access file]

順次編成ファイル、索引順次編成ファイルのように、順次アクセスに適したファイル。(対)直接アクセスファイル

順次処理[sequential processing]

あらかじめ定められた順序で順番に処理していくこと。ひとつの処理が終わってから次の処理を行うため、処理に時間がかかるが、同時に二つの部分を処理しないため、誤りは少ない。逐次処理ともいう。処理の対象は順次編成のファイルのレコードである。

順次ファイル[sequential file] (同)順編成ファイル

順次編成[sequential organization] (参)順編成ファイル

順探索[sequential search] (同)線形探索

順番検査[sequence check]

ファイルのレコードが、特定のキーを基準として、正しい順序関係に保持されているか確認すること。特にデータを複数の媒体で保持している場合、コンピュータで処理する前に並び順の検査が必要である。

順番コード[sequence code]

一連番号をそのまま利用したコード。単純で最も少ない桁数でコード化できる。いったん決めるとコードの追加は末尾へなら可能であるが、挿入はコード変更をともなうため、永続性のある半永久的な固定したデータ、または単なる受け付け順連番などに用いることが多い。

〈例〉JIS都道府県コード

01北海道、02青森県、03岩手県、……、13東京都、14神奈川県、15新潟県、……、46鹿児島県、47沖縄県

順編成ファイル[sequential organization file]

レコードの処理を、先頭から末尾に向かって順次に行う形式のファイル。順ファイル、順次編成ファイルともいう。すべての外部記憶装置に作成可能であるが、典型は磁気テープ上のファイルである。通常、複数個のレコードをまとめたブロックとして扱う。記憶効率がよく、レコード全部一括して順次処理するのに適している。しかし特定レコードへの直接アクセス、書換え、レコードの追加などには難点がある。

順呼出し法[sequential access]

補助記憶装置上に配置してある順(索引、相対ファイルの場合はキーの順)にレコードを呼び出す方法。順編成、索引編成、相対編成のいずれの編成のファイルにも認められた呼出し法である。(対)ランダムアクセス

障害回復[recovery]

システム運用中に障害が発生したとき、その影響を取り除き、もとの正常な状態またはそれに近い状態に戻す作業。復元(reconstitution)、回復

制御、エラーリカバリともいう。障害回復には、障害の発生した機能単位を切り離し処理の続行を可能にするための再構成(バックアップへの切り換え)、バックアップファイルによって誤りの生じる前の状態へもどすロールバック法などがある。データファイルの回復にはアーカイブファイル(定期的な記録)、ジャーナルファイル(個々のアクセスごとの記録)、チェックポイントファイル(主記憶装置の内容を一定時点で記録したファイル)等を利用する。

障害回復テスト[recovery management test]

ハードウェアの故障やソフトウェアのバグなどによって正常な処理が困難となった場合を想定して、障害回復に関する機能をテストすること。

(1)障害の発生状況を監視し、状況を把握できるか。

(2)障害状況から、障害の原因をつきとめられるか。

(3)的確な復旧作業の手順があり、迅速に実施できるか。

条件コード[condition code]

演算や比較などの命令を実行したとき、その結果(2数の大小関係、真偽など)を表すコード。演算命令では演算結果により正、ゼロ、負を示すコードを、比較命令ではオペランドの大小を示すコードをセットする。このほか、オーバフローや桁上げがあったときセットするコードもある。

条件コードレジスタ[condition code register]

算術演算、論理演算、比較演算などを行うごとに、その演算結果によって設定する条件コードを保持するレジスタ。フラグレジスタ、条件レジスタともいう。

詳細設計[detail design]

概要設計をもとに行う入出力帳票や処理手順の具体的な設計。

(同)内部設計

昇順[ascending order]

ソート処理で、配列やテーブルなどの特定キー項目について、小さい順番(英字ならABC順、カナや漢字ならJISコード順)にすること。正順ともいう。表計算ソフトでは、特に指定しなければ、昇順がデフォルト設定されている。大きさをもつ「数値」と文字列としての「数字」では順序が異なることに注意。

{例} 数値の場合　12<21<35<134<226

　　 数字の場合　12<134<21<226<35

仕様書[specification/spec.]

ハードウェアやソフトウェアの仕様、性能などを文書化したもの。仕様書は、対象となる課題や記述法によってさまざまな書き方がある。スペックは略称。一般的にはそのものが示す能力のことを表し、カタログなどに

記載してある性能データをさすことが多い。(類)スペック

小数点[point/radix point/base point]

基数により表現した数(10進数、2進数など)で、整数部と小数部を分離する記号(通例はピリオド)。特に、10進数の小数点を10進小数点、2進数の小数点を2進小数点という。演算装置内では固有の桁を占めないが、表示の際には1桁分を占める。処理系には小数点を文字として扱うもの、数値変数として扱えるものなどがある。

状態語[status word]

処理の結果、ハードウェアの状態などを表すビットで構成した語。

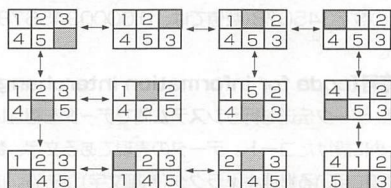
状態語レジスタ[status word register]

CPUや入出力装置などの処理結果や状態を、ビット情報として格納するレジスタ。例えば、プログラム状態語や入出力状態語を記憶する。割込みが発生すると、この状態語を後の処理を再開するために保存する。

状態遷移図[state transition diagram]

情報や状態の時間的順序を規定する図式のひとつ。

①現在の状態でどのような情報を受けたら、次にどのような情報を送り、どの状態に移るかを有向グラフ化した図で、状態をグラフの頂点に、変化の方向を枝(有向、可逆なら双方向または無向)で表す。初期状態と終了状態は必ず存在する。もともとコンピュータの順序回路の設計に利用していたが、汎用性があるためプログラム言語の文法、オペレーティングシステムのタスク(プロセス)制御、データ伝送制御、通信プロセス間の情報の流れなどを正確に説明する手段として広く使用している。

②アルゴリズムの表現において、問題の場面や状況がグラフで表現できるとき、問題の解法を、最小木探索や最短経路探索(枝の重みがどこも同じなら最小手数探索)の問題としてとらえられることが多い。

常駐プログラム[resident program]

コンピュータシステムの動作中、常に主記憶装置内にあるプログラム。必要に応じて補助記憶装置から主記憶装置にロードする非常駐プログラムに対比する語。例えば、仮想記憶システムでは、OSの主記憶監視プログラムは常駐している。

(対)非常駐プログラム

情報[information]

データを集め、比較したり整理したりして、その中から見つけ出したデータに含まれている意味や考察のこと。しばしば、データと情報とを区別なく使用する。データに何らかの処理を行って、意味あるもの、解釈できるものとして得られたものを情報と呼ぶことが多い。情報の最小単位はビットで、二つの状態を示す。いくつかのビットの連なりでさらに多くの状態を示すことができ、文字、数値、各種コードを表現する。これらを連ねてさらに多様なデータや情報を表す。音声や映像などの時間軸や空間軸で変化する量も情報として取り扱う。情報を体系にそって整理したうえで蓄積したものを「知識」という。

情報インフラ[information infrastructure/communication infrastructure]

インフラは「下」を意味する接頭語。インフラストラクチャは、社会の基礎構造、基盤のインフラストラクチャの省略形。情報インフラは情報社会の基盤となる通信網、通信機器などの施設。広義には、その生産体制、供給体制、活用技術やその教育などを含む。

情報落ち誤差[loss of trailing digit]

大きさの著しく異なる二つの浮動小数点数を加算した結果、小さいほうの仮数部の有効桁が失われること。これは、浮動小数点数の仮数部の有効桁が有限桁であることに起因する誤差である。例えば、仮数部の有効桁が8桁であるとすると、

$a=1.2345678\times10^{3}\quad=1234.5678$

$b=1.2345678\times10^{-3}\quad=0.0012345678$

a+b=1234.5690345678となるべきところが、有効桁が8桁のため、コンピュータでは、1234.5690という結果が得られる。すなわちbの値の下位の数字345678(数値では、0.0000345678)は、結果に反映されない。

情報交換用符号[code for information interchange]

情報処理、データ伝送を行うシステム間でデータを相互に利用できるようにするために設けたコード。データの表現である文字、数字、図形文字のほか、制御に用いる機能キャラクタ(制御文字)がある。広く規格化されていることが望ましい。

(例)JISコード、ASCII、ISOコード、JIS情報交換用漢字符号、EUC

情報システム安全対策基準

コンピュータシステムの信頼性、安全性を障害、悪用などから保護するための手段を基準化したもの。簡単に安全対策基準ともいう。

通商産業省は、1975年に設置した電子計算機システム安全対策委員会が他省庁に先駆けて答申した電子計算機システム安全対策基準(1977年)を受け、その後の内容拡充などを経て、1984年に全面的な改定版を発表した。さらに1995年には、クライアントサーバ等の分散処理への対応や地震対策を強化し、名称も「情報システム安全対策基準」に改訂した。安全対策基準は法的な拘束力はもたないが、コンピュータシステムを天災や人災などから守るうえで、コンピュータメーカやコンピュータ室管理の基準となっている。また、警察庁、大蔵省、郵政省、自治省、建設省関係もそれぞれ独自のコンピュータやネットワークの安全対策基準を発表している。

情報処理[information processing]

与えられたデータや情報から目的とする情報を得るために行う、収集、処理、加工、整理などの一連の作業。データ処理と同義であるが、どちらかといえば情報を必要とする人間サイドからみた用語である。

情報処理技術者試験[National Examination for Information Processing Technicians：NEIPT]

情報処理の促進に関する法律第6条に基づいて、1969年から通商産業省が実施している国家試験。情報処理技術者として備えるべき一定水準の能力、技術等を個人に認定することにより情報処理技術者の技能を向上させることなどを目的としている。試験は情報処理技術者の職能、種別に対応した13の区分(1999年現在)があり、毎年4月と10月に分けて実施している。人気は高く、毎回数十万人が受験する。内容はプログラミング技能だけでなく、ハードウェア、ソフトウェアの知識や関連知識にまでおよぶ。

証明機関[certification authority：CA]

暗号化のための公開かぎ証明書の発行、管理、配布を運用する機関。認証局ともいう。なりすまし犯罪を防止するために、ネットワーク上に公開されているかぎが本当に当人、当事者のものかを証明する。

正規の利用者Aは自分の情報と公開かぎを証明機関に登録しておき、証明機関の秘密かぎによるディジタル署名つきの証明書をもらう。Aに通信文を送る利用者Bは、証明機関の公開かぎで利用者Aの情報と公開かぎを得ることができる。A以外の者がAを称して(Aになりすまして)別の公開かぎをネットワーク上に公開しても、そのかぎは証明機関に登録されている正規のかぎではないことを識別できる。

商用データベース[commercial database]

営利目的で情報を第三者に提供するデータベースサービス。現在では電話回線によるパソコン通信でデータベースにアクセスするオンライン方式が普通である。サービスの内容は、政治経済や科学技術などの情報を蓄積

した専門情報データサービスと、一般のニュースや電子メール、電子掲示板などのサービスを中心としたコンシューマデータベースの2種類に分類できる。さらに専門情報データサービスは検索結果が利用者の求める情報そのものであるファクトデータベースと文献の抄録のみのリファレンスデータベースに分けることができる。

シ

上流CASE[upper CASE] (参)CASEツール

初期化①[initialize/clear]

プログラムやサブルーチンの実行に先だって、カウンタやスイッチその他のデータ領域をある値に設定すること。プログラム言語によって、初期化を省略すると値が不定になるもの、デフォルトの初期値(文字属性の領域には空白、またはナルを、数値属性の領域にはゼロ)を設定するものがある。(類)クリアする、(関)初期値

初期化②[media initialize/format]

磁気記憶媒体を(すでにデータを記録してある場合はその内容を消し)新しいファイル用媒体として使用可能な状態にすること。(同)フォーマット

初期値[initial parameter/initial value]

①繰返しループを実行するに先だって、制御変数、領域に設定する定数、条件の値のひとつ。制御変数は、この値をもって繰返しループに入り、終値を越えると繰返し処理を終了する。初期値を入れることを初期値の設定という。

②プログラムの実行に先だってデータ項目に与える値。初期値は、VALUE句やINITIALIZE文などによって与える。初期値を設定していない項目の値は、そのときの主記憶装置の状態によって異なるため、何になるかは特定できない。

ジョブ[job]

ユーザ、オペレータがコンピュータに処理させるひとまとまりの仕事(作業)の単位。複数個の関連をもつジョブステップからなる。

ジョブステップ[job step]

ジョブを構成する単位。ユーザから見たオペレーティングシステムの利用の単位がジョブで、なんらかのデータをシステムに入力し、目的とする結果を得る一連の作業単位である。ジョブは複数の処理プログラムを用い(あるいは複数回用い)、いくつかの段階に分けて実行されるが、その各段階をジョブステップという。ユーザは、JCLにより任意個のジョブステップで組み立てて、ひとまとまりのジョブとして指定できる。

ジョブ制御言語[job control language:JCL] (同)JCL

処理[process/computing]

①データ、データ群に演算を施すこと、関係を作ること。入力したデータ

に対し、あるいはデータ間であらかじめ定めた規則で加工(例えば演算、ソート、マージなど)すること。

②ソフトウェアの開発では、プログラムのアセンブル、コンパイル、編集、ファイルの更新などの操作を広く含む。

③データ処理の工程。これが複数の段階にわたるとき、中核となる処理を主処理、主処理以前に行う処理を前処理、主処理以後に行う処理を後処理という。

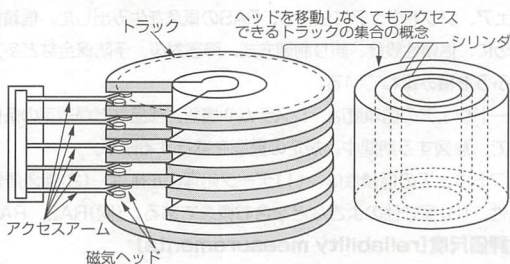
④データ処理システムの特徴的な形態や方式。集中データ処理、分散データ処理、オンラインリアルタイム処理などという。

処理・加工セグメント[processing and manipulating segment]

制御セグメントから呼び出され、引き渡されたデータを処理、加工するセグメント。モジュールのセグメント化を進めるとき、処理条件を明確にした制御セグメントで、実行条件の整理や実行順序の制御を行い、従属モジュールでデータの処理や加工を行う。

処理能力[throughput／performance]　(同)スループット

シリンダ[cylinder]

磁気ディスク記憶装置で、アクセスアームの位置を変えずにデータを取り出すことのできるトラックの集合。各面の同一トラック番号の集まりであり、軸から等距離にある仮想の円筒状に配置できるためシリンダという。磁気ディスクが10枚ある場合、20面のうち最上位と最下位の面は使用せず、1シリンダは18トラックとなる。(関)磁気ディスク

磁気ディスクの構造　　シリンダ概念図

シリンダあふれ域[cylinder overflow area]　(関)索引順次編成

シリンダ索引[cylinder index]　(関)索引順次編成

真[true]

論理変数のとり得る値で、「偽」の対語。ある条件のもとで、命題や述語が成り立つとき真という。(対)偽、(関)論理変数

シングルタスク[single task]

同時にはひとつのプログラムしか実行しないプログラム管理方式。マル

チタスクに対比する語。(対)マルチタスク

シンタックスエラー[syntax error]

ソースプログラムをオブジェクトプログラムに変換する過程で、コンパイラやインタプリタが見つけた文法上のエラー。予約語の綴りミス、オペランドの過不足、構文の矛盾など。コンパイラは通常、エラーの発生する箇所や種類を指摘、出力する。これをエラーメッセージという。(参)論理エラー

診断プログラム[diagnostic program]

電算機の故障時、故障を見つけ状況を報告するプログラム。システムの保全性を高める役割をになう。CPU診断、メモリ診断、I/Oデバイス診断、ソフトウェア診断などのプログラムがある。

伸張[data expansion/data decompression] (参)データ圧縮

シンプレックス構成システム[simplex system]

1台のCPU(中央処理装置)に、最小の構成でファイル装置や通信制御装置を接続したシステム。単一システムともいう。コストは安いが、どれかの機器の障害時には全面停止になるという欠点がある。信頼性は低い。

信頼性・信頼度[reliability]

コンピュータシステムや機器、部品などの機能単位が一定条件のもとで一定期間、安定に実行する能力。コンピュータシステムの信頼性は、ハードウェア、ソフトウェアに依存し、RASの概念を生み出した。信頼性向上のために、信頼性設計、誤り制御方式、障害対策、予防保全など多方面でのくふうを積み重ねている。

①ハードウェアの信頼度は、システムや機器、部品などが規定の条件のもとで、意図する期間中、規定の機能を遂行する確率。

②ソフトウェアの信頼性は、(1)データ処理の正確性、(2)無矛盾性、妥当性、(3)保守性のよさなどを含む概念である。(関)RAS、RASIS

信頼性評価尺度[reliability measurements]

システムの信頼性がどの程度であるかを評価する尺度。個別の機器については、N個の正常なサンプルが規定時間内にr個故障するとき、信頼度は、(N−r)/N で表す。システムは複合体であり、長期的に見てシステムが稼働可能な状態にある時間の割合で可用性(availability)を目安にすることもある。次にあげるものが代表的な尺度である。

(1)平均故障間隔(mean time between failures:MTBF)

システムが正常に動作し続ける(ある故障時から次の故障時までの)時間の平均。平均アップタイムともいう。MTTRと合わせて稼働率や

不稼働率の計算に使用する。

(2) 平均修理時間(mean time to repair：MTTR)

故障が発生した場合に、その修理に要求する平均時間。システムの故障が継続する時間の平均。平均ダウンタイム(mean down time)、平均故障継続時間ともいう。これが大きいことは、システムの信頼性が低いことを意味する。

(3) 稼働率(可用性)(availability)

システム、装置などが正常に動作している時間の割合。稼働率が1に近いほど信頼性が高い。システムの故障が少ないこと、自動回復機能を備えること、修理時間を短くすることなどが可用性を高める。大規模オンラインシステムでは信頼性要求が厳しく、システム設計時、各構成要素の稼働率と構成要素の接続形態からシステム全体の稼働率を予測しておかなければならない。なお、個別構成要素の稼働率は次の式で求める。

稼働率＝ MTBF／(MTBF＋MTTR)

(4) 不稼働率(unavailability)

システムが正常に動作していない時間の割合、または確率。次の式で表す。

不稼働率 ＝ MTTR／(MTBF＋MTTR)

信頼度成長曲線[software reliability growth curve]

テスト終了時を判断する目安のひとつで、テストを繰り返すたびに発見したエラー件数をグラフ化したもの。テストと修正によりシステムの信頼度が高まって行くようすは、ある時点までに発見されたエラー件数の累積に対して、100%に近付いて行く曲線の傾きである程度把握できる。曲線の傾きが水平線に近づく度合で信頼度を判定する。

ス

垂直パリティチェック[vertical parity check]

符号の転送誤りを検出する冗長検査方式の一種で、一文字単位でデータ誤りを検出する方式。送信時には一文字について、パリティビットと呼ぶ冗長符号1ビットを付加する。(対)水平パリティチェック、(関)奇偶検査

水平パリティチェック[horizontal parity check]

符号の転送誤りを検出する冗長検査方式の一種で、データブロック単位に誤りを検出する方式。送信側でデータブロックごとにBCC(ブロックチェックキャラクタ)と呼ぶ冗長符号を付加する。BCCは、ブロック内の

各文字のnビット目をすべて集め、その中で1であるビット数がBCCの対応ビットも含め奇数または偶数になるようなビット列から構成する。

受信側は各文字とBCC対応ビットの1であるビット数を調べて、送信にエラーがなかったかをチェックする。水平パリティチェックも1ビットの誤りしかチェックできないが、垂直パリティチェックを組み合わせることで、2ビット以上の誤り検出ができる確率が高くなる。

ス

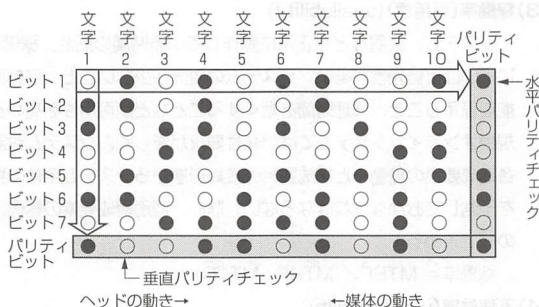



図　垂直パリティチェックと水平パリティチェック

数値データ[numerical value data]

コンピュータ内部で扱うデータの表現を、大きく数値データと非数値データに分けたときの数値データの内容。

(1) 正負符号と小数点を含む数字ワードが表す数の大きさ。

(2) 数値に換算されたデータで、演算や比較の対象になるもの。

数値データの表現は、日常の10進数と計算機内部の演算に適した2進数に分かれる。さらに10進数はゾーン10進数(入出力用)、パック10進数(演算用)に分かれる。2進数は形式によって、固定小数点数と浮動小数点数に分かれる。正負や小数点位置はあらかじめ定めた規則により表現する。(関)基数記数法

(3) 入出力データは数値を意味していても、バイナリデータでなければ情報交換用符号を使って表現した文字である。(関)情報交換用符号

(4) 数字で表現したコードでも、電話番号や10進分類コードは加減算などの演算は無意味なため数値とは考えない。ただし文字などもコード化すれば値をもち、その大小関係を利用して整列、分類、集計などの処理は行うことができる。

(5) 論理データは論理値(真または偽)を表すもので、1と0で表現していても数値ではない。

コンピュータはもともとは数値計算主体であったが、その後記号処理、知識情報処理などが活発になり、非数値データとして画像やイメー

ジなどの扱いも重要になっている。

スーパバイザ[supervisor]　(同)監視プログラム

スーパバイザコール[supervisor call：SVC]　(同)SVC

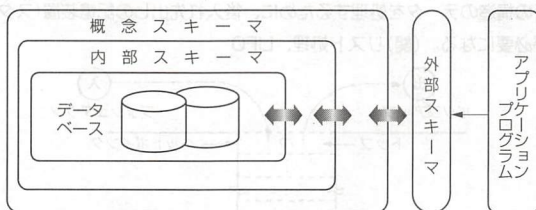
スキーマ[schema]

データベースに蓄積するデータそのものではなく、データの意味や操作のルールの定義(メタデータの集まり)をいう。すなわち、データベースの論理構造、格納構造、物理構造の定義と記述。

代表的なスキーマには、CODASYLのDBTGが提唱した2層スキーマと、ANSI／X3／SPARCが提唱した3層スキーマがある。

(1) **概念スキーマ**：データベースが扱うデータの意味や関係の表現や記述のこと。データベースの応用分野・利用分野の見方や記憶装置に格納する場合の見方とは離れて、人間が認識する意味やデータの構造に近い。この意味で論理スキーマ(logical schema)ともいう。

(2) **外部スキーマ**：データベースの応用分野・利用分野の見地から、データを抽出したり、加工したり、組み合わせたりするための表現や記述のこと。データベースの利用者は、外部スキーマを介してデータベースを利用できる。具体的には、データベースを操作するための記号やデータの指定方法などを含む。

(3) **内部スキーマ**：記憶装置にデータを格納する場合の記述。この意味で物理スキーマ(physical schema)ともいう。内部スキーマは、蓄積したデータの量や性質、利用目的によって変更されるが、他のスキーマとは独立に定義されていれば利用者には影響がなく、データの独立性が保たれる。

スキップ[skip]

ある命令や処理を意図的に省略したり、抜かしたりすること。処理するデータと処理しないデータを選択する場合、エラーを回避して処理を実行する場合、条件コードを判断して飛越し先を変えること。

スクリーンエディタ[screen editor]　(参)エディタ

スクリプト[script]

定型化された手順を記述した命令群。プログラム言語によるプログラム

とは異なり、マクロ言語やスクリプト言語のようなコマンドで表現する命令。定型的処理の自動運転を行うMS-DOSのバッチプログラムやUNIX系OSのスクリプトファイル、マクロ機能のあるアプリケーションのマクロ命令の記述などをさす。

スクリプト言語[script language]

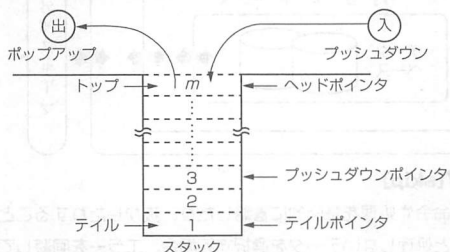
ある目的のスクリプトを記述する言語。一般的にテキストで記述する。

スケジューリング[scheduling]

システム資源を有効に活用するために、各装置を利用する時間を適正に配分すること、またはそのためのオペレーティングシステムのプログラム機能(スケジューラという)。ユーザはコンピュータシステムに多くのジョブを与える際に、使用する資源や優先順位を指定する。OSはコンピュータ資源を遊ばせないようにスケジューリングし、オペレータに対しファイルのローディングなど必要な処置をメッセージで要請する。この機能で数多くのジョブの連続運転が可能になる。

スタック[stack]

後入れ先出し(LIFO)のデータ構造。後入れ先出し記憶装置または記憶領域をさすこともある。複数のデータを順次に記憶するが、取り出せるデータは常に1件で、最も後から記憶したデータのみである。列の一端だけで追加や取り出しが可能である。新しく記憶するデータを古いデータに積み重ねることをプッシュダウン(push down)、または単にプッシュ(押し込み)という。逆にスタックから最も新しいデータを取り出すことをポップアップ(pop up)、または単にポップ(跳ね上げ)という。

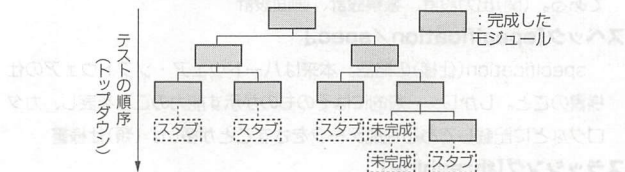
例えば、サブルーチンの戻り番地の管理、多重の割込みを処理する入れ子の構造のデータを処理するために、後入れ先出しの記憶装置(スタック)が必要になる。(関)リスト処理、LIFO

スタックポインタ[stack pointer：SP]

配列でスタックを実現するとき、アクセスする場所を指定するポインタ。記憶場所を示すアドレス、または添字であり、特に記憶構造がスタックの場合、こう呼ぶ。スタックを配列で実現するとき、スタックポインタ

になるアクセスアドレスを特にヘッドポインタ、スタックの末尾のアドレスをテールポインタという。

スタブ[stub]

①上位レベルのモジュールまたはモジュールの集まりをテストするために、下位レベルの未完成モジュールのインタフェース部分だけを設定したテスト用のプログラム。(関)トップダウンテスト

②RPC(遠隔手続き呼出し)で、あるプログラムが別のプログラム中の手続きを呼び出すとき、引数と結果を授受するモジュール。

スタンダード[standard]

標準。ものごとを決める際の規範、まもるべき基準。例えば、各種製品の互換性を維持し、生産効率、流通効率を高めるために、標準化機関が定めた規格。また市場占有率が高く、事実上標準に見なされる有力メーカの規格。

ストア[store]

①レコードをファイルに書き込むことなど、データの記憶装置への格納。反対に記憶装置からデータを取り出すことをロードという。
(同)アンロード、(対)ロード

②アセンブラ命令のストアは、レジスタの内容をメモリへ格納すること。

スナップショットダンプ[snapshot dump]　(関)ダンプ

スパイラルモデル[spiral model]

ウォータフォールモデルとプロトタイプモデルの両方の手法を折衷したモデル。開発の各期階で必要とする技術者の技能が異なるため、ウォータフォールモデルでは初期と後期にシステムアナリストやシステムエンジニアが不足し、中期にプログラマが不足する現象が起きるが、これを解消するため、開発の初期から独立性の高いモジュールについては設計、プログラミング、テストを行い、順次にシステムを開発する手法。

スプール[spool／simultaneous peripheral operations on line]

資源の効率的運用を図ることを目的に、中央処理装置のジョブと入出力処理を並行して行うこと。読取り装置やプリンタ等の周辺装置は中央処理装置に比べて処理速度が遅いため、周辺装置を直接使用すると中央

処理装置に処理待ちが生じる可能性がある。そこで、中央処理装置と周辺装置の間でデータを転送するとき、補助記憶装置を緩衝記憶として用い、処理の遅れを短縮させる。周辺装置は補助記憶装置からデータを受けとる。(参)バッファ記憶装置

スペーシングチャート[spacing chart]

コンピュータの処理結果をディスプレイ画面や用紙に出力する場合に、どの位置にどのデータを何桁で印字するかなど、形式を設計するために用いる用紙。1画面、帳票1ページの文字数、行数に合わせた罫線が印刷してある。(関)出力設計、帳票設計、画面設計

スペック[specification／spec.]

specification(仕様)の略語。本来はハードウェア・ソフトウェアの仕様書のこと。しかし、一般的にはそのものが示す能力のことを表し、カタログなどに記載してある性能データをさすことが多い。(類)仕様書

スラッシング[thrashing]

仮想記憶方式で、非常に頻繁にページングを行うこと。システムで使用可能な実記憶領域に比して、仮想記憶上のプログラムが大きい場合、またマルチプログラミングの多重度が高い場合、使用可能な実記憶容量が少なくなるとスラッシングが発生し、システム全体の効率が悪くなる。

スループット[throughput]

①単位時間当たり処理できる仕事量。高価なコンピュータ資源を効率よく使用するため、オペレーティングシステムでは、スループットの向上が主要な目標になる。この実現のため、マルチプログラミング技法、スプール技法などがある。

②データ転送の効率(単位時間当たりのデータ通信量)の意味でも、用いられているようになっている。

スロット[slot]

①データを格納する論理的な単位。

②仮想記憶システムで、外部ページング記憶装置上の、各ページを収容する単位。ページと同じ4キロバイトのブロックにフォーマットする。

③拡張ボードやPCカードなどを収納するための差し込み口。ドライブなどの収納口はベイという。

スワップ／スワッピング[swap／swapping]

①主記憶装置の領域の内容と補助記憶装置の領域の内容を交換する処理。OSは、このスワップという手法を利用してシステムが実際に搭載しているメモリ容量の制約を越える大きなプログラムやデータを扱う。この手法を仮想記憶という。

(類)ロールイン・ロールアウト

②プログラミングでは、2か所の記憶領域の内容を互いに入れ換えること。例えば、ソートの必要からA番地とB番地のデータを交換すること。

正規化[normalization]

ある量を定められた範囲内に収まるよう調整すること。

①浮動小数点表示で、ある数の小数部があらかじめ定められた範囲内に入るように指数部と小数部を調整すること。例えば、10進数で小数部xの範囲を $1>x \geqq 0.1$（xは1以下の最大値）に定めてある計算機で演算結果として 0.00632×10^8 を得たときは、これを 0.632×10^6 に変更する。

②関係データベースで、データの独立性を保つために行うデータのグルーピングのこと。関係データベースでは、重複データの排除、データの更新、挿入、削除の影響の排除などデータの独立性の維持が重要であり、この条件を満たすために次のようなグルーピングを行う。

(1)繰り返しのある項目の独立化（第1正規化）

(2)複数のキー項目が揃ってから初めて意味がある項目以外の独立化（第2正規化）

(3)他の項目からの間接演算で同じ内容が得られる項目の除去（第3正規化）

正規形[normal form]

関係データベースで正規化をしたデータ。

(1)第1正規形(first normal form)：あるデータベースを構成するレコードが、それ以上分解できないところまで分けられているとき、その表は第1正規形という。すなわち、ファイルでいえば繰り返しのある項目がなくなりすべて独立した項目の表現になる。

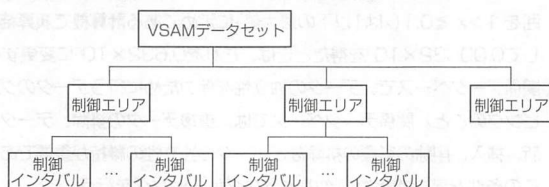
(2)第2正規形(second normal form)：あるデータ項目(a)が決まると、他のデータ項目(b)が一意に決まるとき(bはaに機能従属するという)、必要最小限のデータの組合せになるように表を分割すること。表に機能従属がないとき、第2正規形という。

(3)第3正規形(third normal form)：あるデータ項目(a)が決まると、他のデータ項目(w)が決まり、項目(w)が決まると、他のデータ項目(b)が決まるとき(bはaに推移従属するという)、食い違いが生じないように表を分割すること。表に推移従属がないとき、第3正規形という。

制御インタバル[control interval：CI]

VSAMデータセットを構成する論理的な構成要素のひとつであり、制御エリア(CA)を細分化した領域。制御インタバル(CI)がいくつか集まって

制御エリアを構成する。ひとつのCIには、複数の論理レコードが記憶され、CIはVSAMにおける入出力操作の単位となる。ひとつのデータセットを構成する制御インタバルは固定長であるが、これを構成する論理レコードは固定長の場合も可変長の場合もある。

制御インタバルには、レコード定義フィールド(record define field：RDF)と制御インタバル定義フィールド(control interval define field：CIDF)がある。RDFは論理レコードを記録するフィールドであり、CIDFは空きスペースの位置と大きさおよび各論理レコードの長さと制御情報を記録するフィールドである。それぞれのフィールドは固定された位置にあるのではなく、制御インタバルの前と後から作成されていく。

制御エリア[control area：CA]

VSAMデータセットを構成する論理的な区域。レコードを記憶する。複数の制御インタバルによって構成される。また、複数のCAが集まってひとつのデータセットを構成する。将来のレコード追加に備え、VSAMデータセットの終わりには未使用の制御エリアを用意する。これを空き制御エリア(空きCA)といい、空きCAが無くなった場合は、新しいCAを設定することができる。

制御コード[control code]　(同)機能キャラクタ

制御セグメント[control segment]

ひとつのモジュールの中でも、いくつかの機能部分に分けて考えるとき、細分化した単位をセグメントという。上位のセグメントは、モジュール全体の管理を行うだけの場合、特に制御セグメントという。制御セグメントは従属セグメントを呼び出すだけであり、実際のデータの処理・加工は従属セグメントで行う。

制御プログラム[control program]

コンピュータをむだなく有効に利用するために、コンピュータ全体の作業の流れを監視し制御するプログラム。処理プログラムとともに、オペレーティングシステム(OS)を構成する重要なプログラムである。

制御文字[control character]　(同)機能キャラクタ

静止画データ[image data]

写真、絵、図形などをディジタル化したデータ。画像データのファイル

は、ファイル容量と処理時間の問題から、いくつかの圧縮・伸張方式が提案されており、複数のファイルフォーマットが存在する。カラーの静止画の解像度をあげると、1枚の静止画でも大きなデータ量となるため、圧縮して保管するのが一般的である。フルカラー静止画データの圧縮技術の標準としてJPEGがある。そのほか、GIFもインターネットのWebページの静止画データとして多く利用されている。(参)JPEG、GIF

整数演算[integer arithmetic operation]

整数型のデータ形式で行う演算。小数部は扱わない。10/3を3とするような小数部無視の演算で、浮動小数点演算などで問題になる桁落ちはないが、扱える数値の範囲は狭い。

整数化[round]

①実数型から整数型への型変換。小数点以下の値を切り捨てる。また整数演算をする場合、FortranやCOBOLでは結果の小数点以下を切り捨てる。

②小数から整数にすること。一般的には小数点以下の四捨五入、切上げ、切捨てなどの方法がある。(類)丸め

整数型[integer type/int]

データ型のひとつ。整数は、実数のうち、小数点以下をもたない数のこと。プログラム言語によって、正負の符号の扱い、有効けた数の扱いが異なる。

性能管理[performance analysis]

システムの性能は、ユーザやデータ量の増加などで低下することがあるので、稼働中の性能を監視、分析し、信頼性を保持すること。将来の処理量予測、性能向上のための現状評価や問題点の探索を含む。

性能テスト[performance test]

開発したコンピュータシステムが所定の処理能力を発揮するか検査すること。システムにより、処理能力の評価基準は異なるが、通常は単位時間当たりに処理できる仕事量、入力から出力を得るまでの応答時間、過大なデータを与えた場合の遅れなどが設計基準を満たすか、運用上の問題になるかを調べる。

西暦2000年問題

西暦年月日表現をyymmddのように、年部分を下2桁の数字で済ませたことが西暦2000年以後のコンピュータシステム運用に悪影響を及ぼすこと。クロック問題ともいう。システムが、例えば暦年数字02を2002年と判断しないで1902年と判断する場合に生ずる諸問題である。西暦年を4桁表現に変更するか、区別する入出力ルーチンを追加するなどの方策をとる。既に利用しているプログラム上の問題箇所を検索するツールなども販売されている。

整列[sort／sorting]　(同)ソート

セーブ[save]

　一般的には特定の情報を破壊や消滅から守るために別の記憶媒体、別の場所に複写(または移動)すること。退避ともいう。特に主記憶装置内のプログラムやデータを外部記憶装置に記憶するときに用いる語。

セ

セキュリティ[security／secret protection]

　コンピュータ資源を破壊、障害、盗用、悪用などの犯罪等から守ること。コンピュータ犯罪の防止やプライバシ保護のために、アクセス権をもたない者からプログラムやデータを隔てること。コンピュータセキュリティ、安全性ともいう。主として、ハードウェア機能の破壊(事故、災害による)、ソフトウェア機能の破壊、侵害(運用ミス、負荷集中等による)のほか、データの不正使用、破壊、改ざん(偽情報の混入)等の悪用に対する抑制、防止対策、その機構をさす。

(1)アクセス者の正当性を確認する認証技術

(2)データそのものをアクセス者から隔離する技術

(3)アクセス者を特定できる監視技術

　具体的には、ハードウェアを火災、出水、埃などから守る設備、IDカード設備の整備、パスワード、コピープロテクト、コールバック、データの暗号化、オペレーションログの実施などをセキュリティレベルに応じて行わなければならない。これに加え、システムダウンにそなえるバックアップ技術や回復技術も重要な安全性確立の技術である。

セクタ[sector]

　磁気ディスク、ディスクパック、フロッピーディスクの各記憶媒体は、記憶面を複数のトラックに分割し、任意のトラックをアクセスできるようにしてある。その分割したトラックをさらに分割した最小単位をセクタという。トラックは等分割する場合や任意のサイズに分割する場合がある。大きなデータは複数のセクタにまたがって記録し、小さなデータは1セクタに記録する。しかし、すでに記録したデータがあると、他のデータをそのセクタに記録できないため無駄が生ずることがある。(参)トラック

セグメント[segment]

　プログラムやデータをいくつかの部分に分割し、それ自体で完結した処理ができる、あるいは処理の対象となるまとまりのこと。

　仮想記憶方式やマルチプログラミング方式で記憶容量に比して大きなプログラムやデータ、複数のプログラムを扱うときに利用する。仮想記憶方式では、記憶領域をページと呼ぶある一定の大きさに分割し、ページ単位に主記憶上にロード(ページイン)、アンロード(ページアウト)する。このページを複数個まとめたものをセグメントという。プログラム全体が主記

憶上に存在しなくても実行できるように、プログラムの構成要素ごとに定義する。セグメンテーションは、セグメントに分割して管理すること。(関)仮想記憶、セグメント化

セグメント化[segmentation]

分割したプログラムモジュールの内部は、さらに幾つかの機能部分に分けて考えることができる。モジュールを細分化した部分をセグメントという。セグメントは、論理的にまとまりのあるステートメント(文)の集まりである。モジュールをセグメントに分割することをセグメント化という。

節[node]

木構造で、枝(branch)の集合点を節(node)という。したがって節は枝の分岐点(節点)になる。グラフ理論では点、通信ネットワークでは、局ということがある。(同)ノード、(参)木構造

絶対アドレス[absolute address] (参)実効アドレス

セットアップ[setup]

新規のハードウェア構成要素を互いに接続し適切な設定を行い、記憶媒体の初期化、基本ソフトウェアのインストール、アプリケーションソフトのカスタマイズなどを行う作業。次のような作業の全体。

(1) 各装置に必要な部品要素(本体、インタフェース、ケーブルなど)やソフトウェア(デバイスドライバ、組込み用ソフト、マニュアル、利用者登録表など)の確認。
(2) 増設メモリや増設インタフェースボードなどの本体内部への取り付け。
(3) 周辺機器と本体のケーブルの接続、各種スイッチの設定、電源の投入。
(4) オペレーティングシステムのインストール。
(5) 装置のドライバのインストール。外部記憶媒体の初期化、領域確保。
(6) アプリケーションプログラムのインストール、カスタマイズ。
(7) コンピュータシステム全体のシステム環境設定。

セプテット[septet]

7ビットで表現したコード。7ビットバイトともいう。ASCII符号では32個の制御文字、96個の大小英文字、数字、特殊文字を定義している。またJISでは情報交換用符号をJIS-X0201-1976で定めており、ローマ文字用7単位符号、片仮名用7単位符号、ローマ文字と片仮名用8単位符号の3種類がある。7ビットでは最大128(＝2^7)種類の文字を表現できる。

セルフアセンブル[self assemble] (関)アセンブル

セルフコンパイラ[self compiler] (対)クロスコンパイラ

ゼロクリア[zero clear]

累算(累加＝足し込み)を行うとき、累算結果を記憶する変数の値を最初に0にしておくこと。プログラムではあらかじめ初期値としてゼロクリア

する場合と累算の実行直前に0を代入する場合がある。

ゼロ除算[division by zero]

数値演算が禁じているゼロでの除算。その結果生ずる内部割込み。そのとき表示するエラーメッセージ。

ゼロ抑制[zero suppression]

数字の表現を見やすくするために、小数点以上(整数も含め)の有効数字(有効けた)より上位にある0をなくしたり、*で置き換えたりすること。ゼロ消却ともいう。例えば「0012.3」を「12.3」と上位の「00」を消して表示すること。例えばCOBOLでは、PICTURE句の文字列が「Z」のときはゼロが空白に、「*」のときは「*」に置き換わる。

全角文字[full size character]

ディスプレイ、プリンタなどが表示、印字する標準サイズの文字で、縦横寸法がほぼ等しい文字の通称。通常はJIS情報交換用漢字符号で定義した文字。漢字と非漢字(記号、英数字、ひらがな、カタカナ)などの2バイト文字。

線形探索[linear search/sequential search]

配列、またはリストにあるデータを一端から順に探索する技法。配列(リスト)中に、レコード以外のデータ(例えば、空欄やEOF)があるか、レコードの件数は既知か、同じキーのレコードが多重登録されているかなどでアルゴリズムに若干の違いがある。データ量がn倍になれば、探索のためのデータ量もn倍になる。逐次探索、順次探索ともいう。(関)番兵

線形リスト[linear list]　(関)リスト構造

選択①[select]　(参)基本制御構造

選択②[select/selection]　(関)関係代数

選択ソート[selection sort]

簡単な内部ソートのアルゴリズムのひとつ。配列中のデータの最小値(または最大値)を見つけ配列の一端におき、探索範囲を1件分狭め、同様の操作を繰り返すソート法。バブルソートに比べ交換の回数は減るが、やはりデータ量の2乗に比例する計算量が必要で、すぐれたアルゴリズムとはいえない。最小値法(最大値法)ともいう。

ソ

増加テスト[incremental test]

テスト済みのモジュールに未テストのモジュールを結合して、順次に行うテスト。1回のテストで検証されるのは、未テストモジュール内部か結

合部分ということで、比較的エラーの発生部分の特定がしやすい。テストが終了、修正したら、次のモジュールを加えて同様のテストを繰り返す。モジュールの結合の順序によりトップダウン型とボトムアップ型に大別できる。

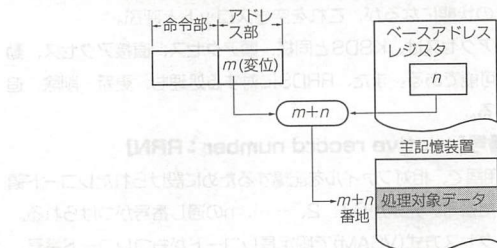
総合テスト[integrated test／system test]

開発されたシステムが、総合的に見てシステム設計段階の目的・目標を実現しているかを確認するテスト。システムテストともいう。主なテスト項目は、機能テスト・過負荷テスト・安定性テスト・耐障害テスト・性能テストなどであり、運用の現場にシステムが移行された場合を想定して開発者が行うテストである。

相対アドレス[relative address]

別に指定したアドレスを基準として、相対的に表したアドレス。多重プログラミングで、いくつかのプログラムを同時並行処理する場合、プログラムの記憶領域を絶対アドレスとして固定すると、記憶装置内に空きの領域があってもほかのプログラムを実行できない場合がある。そこで、実記憶装置に格納する段階で相対アドレスから絶対アドレスに修飾(変更)する方法をとることにより、空いている領域を有効に利用し、多重プログラミングを可能とした。

相対アドレス指定[relative addressing]

命令語のアドレス部に基準値(ベースレジスタ、または命令アドレスレジスタが記憶する値)からの変位を指定する方式。プログラムを記憶装置のどこにでも置ける点が便利。(参)再配置可能プログラム

相対バイトアドレス[relative byte address：RBA]

VSAMデータセットで、各レコードの記憶アドレスを表現するアドレス。データセットの最初の制御エリア(CA)の最初の制御インタバル(CI)の先頭番地を基準0として、基準から数えたバイト数でレコードの所在を表す。このようにCI別に相対的に表したアドレスを用いると、シリンダ番号やトラック番号で指定する必要がなくなりハードウェアからの独立性が増し、データセットの保守作業やレコードの移動が容易になる。

相対ファイル[relative file]

相対レコード番号によりファイル内のレコードを識別できるように編成されたファイル。相対レコード番号の昇順に呼び出す順呼出しと、乱呼出しが可能である。

相対編成[relative organization]

ファイル編成法のひとつ。ファイル上の最初のブロックを0番ブロック、次を1番ブロックとし、そのブロック番号を指定してファイルを読み書きできる。例えば、あるファイルの先頭から数えて58番目のブロックを読み出すには、相対ブロック番号を58と指定すると、ソフトウェアがブロックの位置を算出し、該当トラックおよびブロックにヘッドを位置づける。このような乱呼出しができるファイル編成。通常1ブロック1レコード(固定長レコード)であり、この番号を相対レコード番号という。

相対編成ファイル[relative organization file]

(同)相対ファイル

相対レコードデータセット[relative record data set : RRDS]

直接編成ファイルと同様に、レコードのキーをレコードアドレスに変換して、レコードへの直接アクセスが可能なデータセット。RRDSでは、データコンポーネント内はスロットという単位に分解される。スロットは論理レコードを記憶するための器であり固定長である。スロットには先頭から順に番号が付けられ、この番号でレコードの位置を特定する。この番号を相対レコード番号という。論理レコードはキー値と対応するRRNのスロットに記録される。このとき、対応するキー値をもつレコードのないスロットは空きの状態になるが、これを空きスロットと呼ぶ。

RRDSへのアクセスは、KSDSと同様、順アクセス、直接アクセス、動的アクセスが可能である。また、RRDSに対する処理も、更新・削除・追加が可能である。

相対レコード番号[relative record number : RRN]

①COBOLの用語で、相対ファイルを記憶するために設けられたレコード領域にふられた番号。先頭から1、2、……、nの通し番号がつけられる。

②仮想記憶アクセス方式(VSAM)で固定長レコードがもつレコード番号。VSAMファイル内の相対編成ファイルに相当するファイルの最初のレコードからの隔たりで、物理的なディスクのアドレスを計算できる。

挿入[insert/supplement/addition/insertion]

データの並び(文字列、配列、レコードの並び、ファイルなどを含む)の中に別のデータの並びを適切な位置にはさみ込むこと、あるいはその機能。追加ともいう。表計算ソフトでの列や行の挿入は、システムが自動的に処理する。プログラムで、ファイル中のレコードの並びに新たなレコー

ドを挿入する場合は、挿入レコード以降のレコードを後方に転記したり、レコードの並び方を指定するポインタの値を書き換える必要がある。

双方向リスト[bi-directional list]　(関)リスト構造

添字[subscript]

配列やテーブル中の特定の要素を識別するために配列名やテーブル名とともに用いる順序数。プログラムの中では、整数または変数の順序数要素をかっこでくくって表す。(参)配列

ソースデータ[source data]

データ処理の対象となるデータの最初の形。原始データともいう。例えば、取引が発生で生じたデータは、当初は伝票に記録し、後に処理する。この伝票が原始データである。一般に原始データはそのままの形式ではコンピュータ処理に適した形式になっていないため、入力に際して入力変換を要する。

ソースプログラム[source program／source code]

プログラマがプログラム言語で表現したプログラム、またはそのテキストファイル。例えば、C、C++、Fortran、COBOL、アセンブラなどで記述したプログラム。原始プログラムともいう。コンピュータがこのプログラムを実行するためには、さらに言語翻訳プログラム、連係編集プログラムにより機械語に変換する必要がある。(関)コンパイル、アセンブル

ソート[sort]

ファイルまたはレコード中の各データ項目を定められた規準(例えば五十音順、数の大小、日付順など)に従って順序よく並べ換えること。整列、分類ともいう。順序には、昇順(正順)とその逆の降順(逆順)とがある。

並べ換えにより主処理を正しく能率よく行うことができる。そのため前処理としてデータをソートすることが多い。また出力データもソートにより順位がわかる、検索が容易などの理由でソートすることが多い。

実際のソートには、プログラム言語やアプリケーションソフトに備わっているソート命令やソートコマンド、専用のソートプログラムを使用する場合が多い。この場合は入出力ファイルとキー項目の簡単な定義をするだけでよい。ソートの際に大量のレコードを短時間に処理するために、アルゴリズムがくふうされ、各種の手法がある。

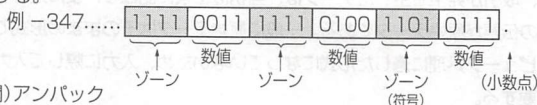
(関)外部ソート、内部ソート、ソートキー、基本交換法

ソートキー[sort key]

各データ項目を順序よく並べ換えるために、指定する項目。通常のレコードは、複数の項目からなり、どの項目で並べ替えるかにより出力が異なる。また複数のソートキーの優先順位を指定できるのが普通である。社員ファイルで、入社年度をソートキーにすれば社歴の順に、生年月日を

ソートキーにすれば年齢順に、氏名の読みをソートキーにすれば五十音順に並べる。

ゾーン形式[zone format]

ゾーン10進数の表現形式。1バイトで1桁の数字を表し、小数点の位置をLSBの右と想定し、符号は最下位バイトの上位4ビットにおく。この形式は、各桁を1バイトで表し、各バイトは上位4ビットのゾーン(zone)部(EBCDIC系は1111、JIS系は0011)と下位4ビットの2進化10進表記法(BCDコード)からなる。符号は最下位のゾーン部で示す(通常、プラスは1100、マイナスは1101)。パック形式に比べおよそ2倍の領域を必要とする。

(関)アンパック

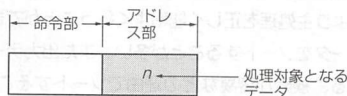
ゾーン10進数[zone format]

計算機内での10進数値の表し方のひとつ。入出力装置(例えばラインプリンタ)は通常この形式でデータのやりとりを行っている。アンパック形式、外部10進ともいう。(関)ゾーン形式

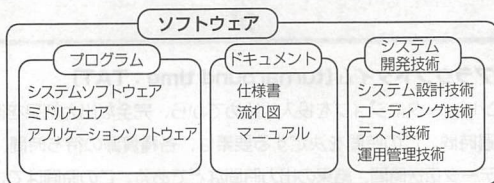
属性[attribute] (同)アトリビュート

即値アドレス指定[immediate addressing/zero level addressing]

命令のアドレス部の内容が、データを格納してある主記憶装置のアドレスではなく、処理の対象となるデータの値そのものであるという指定方式。ゼロレベルアドレッシングともいう。

命令部
アドレス部
n
処理対象となるデータ

ソフトウェア[software:S/W]

不特定多数の人が頻繁に使用するプログラム。一般には大規模で複雑になることが多く、複数人で開発する場合が多い。広義には、プログラム、プログラム開発技法、コンピュータの運用法、利用技術やこれらに関する書類(ドキュメント)などを抽象的にとらえたものである。狭義には、プログラムやマニュアルそのものをさすことが多い。

ソフトウェアはハードウェアと比べ、論理的に複雑な表現ができ、改変しやすく、複製ができるため、データ処理システムの中では工業製品であるハードウェアを多様な目的に利用できるようにする役割をになう。ソフトウェアは、その役割から、(1)システムソフトウェア、(2)ミドルウェア、(3)応用ソフトウェアに分類できる。

ソフトウェアパッケージ[software package]

注文生産によるソフトではなく、ある機種に合わせて多くのユーザが類似の目的で使えるように考慮して商品化したソフト。大量販売のねらいがあるため廉価で多機能なソフトが多い。大型汎用コンピュータの個別アプリケーションソフトはほとんどが注文生産である。ユーザの圧倒的に多いオフコンやパソコンでは、パッケージソフトが大量に普及している。大局的に見れば、多くのユーザがソフトウェアパッケージを利用すれば、ユーザごとに類似のソフトウェアを開発するという重複の無駄がなくなる。また、大量販売によりパッケージの開発単価が下がり、需要が増える経済効果が期待できる。

ソフトウェアパッケージはその内容により、さらに分類することがある。

(1) **アプリケーションパッケージ**：給与計算、顧客管理、在庫管理などのデータ処理用のプログラム

(2) **システムパッケージ**：オペレーティングシステム、日本語FEP、表計算ソフトなどの基本ソフトウェア

(3) **ツールパッケージ**：システム開発支援、システム運用支援、ユーティリティなどのミドルウェア

ソフトウェア品質[software quality]

広義には、ソフトウェア機能、ユーザの操作性、開発者の効率、保守性のよさ、移植性をさす概念。狭義には、運用時の信頼性をさす。ソフトウェア品質の向上のためには、エラーを作らないこと、作ったエラーを発見し除去することが必要で、綿密な設計技法・テスト技法の徹底、品質評価項目と基準の設定が肝要である。

ソフトコピー[soft copy]

ディスプレイのような一時的な表示装置に文字や図形などが表示されたもの。機械的動作の制約を受けないことから、表示スピードや表示方法などの面で利点がある。(対)ハードコピー

タ

ターンアラウンドタイム[turnaround time：TAT]

コンピュータにジョブを投入し始めてから、完全な出力結果を得るまでの経過時間。この時間を決定する要素は、各種資源の待ち時間、処理時間、データ伝送時間、結果の出力時間などである。この時間はできるだけ短いほうがよい。オペレーティングシステムが発達した理由のひとつが、この時間の短縮にある。(参)応答時間

大域変数[global variable]

プログラムで、各ブロックを超えて、同一名称で使用できる変数のこと。Fortranでは、COMMON文で定義する。Cでは、変数をEXTERNAL宣言して大域変数として扱う。(対)局所変数

タイムシェアリングシステム[time sharing system：TSS]

コンピュータの共同利用の一形態であり、マルチプログラミング技法の一形態。ネットワークで接続した複数のユーザ端末が、独自にホストコンピュータと会話する形で処理を行うシステム。時分割システムともいう。

ホストコンピュータは、各端末のプログラムに使用時間を小さく区切り(微小時間をタイムスライスという)、順番に割り当てて実行する。コンピュータは多数のプログラムを同時並行的な処理をしているように見える。そのため、ホストコンピュータとセションを確立した使用者は、あたかもコンピュータを独占しているかのような感覚で使用できるという特色がある。

|←→| タイムスライス　→ 時間

CPU	ジョブ										端末
CPU	Aのジョブ	実行			実行			実行			Aさん
	Bのジョブ		実行			実行			実行		Bさん
	Cのジョブ			実行			実行			実行	Cさん

(関)ラウンドロビン

第四世代言語[fourth generation language：4GL]

COBOL、Fortran、PL／Iなどの高水準言語(第三世代言語)よりさらに抽象度の高い命令をもつプログラム言語。実際には汎用言語ではなく、特定の目的をもつ言語が多い。言語の世代区分はハードウェアのように明確ではない。第四世代言語は、ソフトの開発、保守の生産性を向上させ、コストを大幅に減らすことをめざしている。エンドユーザ向けの表計算ソフトのマクロ機能やデータベースアクセス言語(DML)、ビジュアルプログラミング用言語など広い範囲のものを指す。

対話型処理[conversational processing]

広義には、コンピュータシステムとデータや制御情報を交換しながら進

める処理。狭義には、人間がコンピュータシステムに次々と指示を与えながら進めるデータ処理。会話型処理ともいう。すなわち、コンピュータシステムの操作の一形態で、利用者とシステムとの間で交互に取り交わす投入(入力)および応答(出力)に基づいてデータ処理を行う方式。典型的な例としては、パソコンのワープロ操作、表計算ソフトの操作などがあげられる。機能の選択、データの入力、結果の出力などは、人間の判断と操作を加えながら進展する。

タグ[tag]

①レコードなどのある項目を示すために、それに付けたひとつ以上の文字。

②データの性質などを示すために、データの前または後に付ける文字。受信データの誤りを検出した時などに表示する目印など。標識ともいう。

③文書や文字の体裁を定義する目的で、本文テキスト中に組み込む情報。一般に文書の体裁(ページ当たりの字数や行数、ヘッダやフッタの扱い)や文字の属性(書体や大きさ、色、文字の位置など)は、次の方式で扱う。

(1)本文(テキスト)と別のファイルに文字属性情報や文書の体裁を記録し、出力時に参照する方式では、1文書が複数のファイルとなる。

(2)本文中に制御コードとして組み込み、出力時に体裁を整える方式では、1文書が1ファイルになり、通信にはこのほうが適している。

(関)HTML

④文章のある部分に対する詳しい説明が別の場所にある場合、その場所を記録し参照可能にしたもの。HTMLやSGMLのようなタグ付け可能なマークアップ言語や高性能ワープロソフトでは、タグ付けすることによってハイパーテキストを実現する。普通、表示された文章でタグ付けされた部分は、他の部分と異なる色となっていたり、下線が引かれていたりする。そこをクリックすると、詳細記述やイメージが表示されたり、音声が出力されたりする。

タスク[task]

オペレーティングシステムの制御プログラムが管理するプログラムの単位。プロセスともいう。通常、ひとつの実行文から次の実行文もしくはジョブ文までをひとつの単位とする。また、これらのタスク群を管理することをタスク管理といい、制御プログラムの機能のひとつである。

妥当性検査[validity check]

データや命令が論理的に妥当であるか、処理に適した形式であるかをチェックすること。妥当性検査には書式検査、件数検査、限度検査、冗長検査を含む。

タブ[tab／tab marker／tab code]

印刷や表示の書式を整えるためにカーソルを同じ行の指定位置までずら

す制御文字、区切り文字。または、それを実現するための機構。パソコンやワープロなどでは8文字単位に行頭を揃えるものが多い。ファイル内部では1文字(タブコード)として扱うが、画面や用紙に表示、印字する時は、プログラムで必要な空白を補って出力するため、行頭が揃う。タブコードの入力にはキーボード上の[TAB]キー、[→|]キーを用いる。

タプル[tuple] (同)組

ダミーレコード[dummy record]

記憶領域を占有するための実際のデータではない便宜的な空のレコード。直接編成ファイルのレコードはキー順ではなく磁気ディスクのアドレスに不連続に記録されるが、レコードの欠番部分を埋めるために使用する。

(1)直接アドレス方式では、アドレスごとに順次に実在レコードかダミーレコードを書き込んで行く。レコードがなくなれば残りアドレスにダミーレコードを書く。

(2)間接アドレス方式では、まず全領域にダミーレコードを書き込んで行く。その後実在レコードにより算出したアドレスに実際のレコードを書く。

単一ボリュームファイル[single volume file] (関)ボリューム

段階的詳細化[stepwise refinement]

構造化プログラミング技法のひとつ。プログラム作成の課題をまず抽象的にとらえデータと処理の内容を記述し、次の段階ではその記述をより具体化したデータと処理の記述に分解(break down)する。さらに次の段階でプログラム言語に近い形式でデータと処理の記述を行う。このような具体化と詳細化を繰り返し、最終的にプログラム言語による記述を完成する。

この技法によれば、処理の流れを人間的な思考で定義し、次第にコンピュータ処理に準じた形式に変換していくことになり、誤りが少なく、また読みやすいプログラムが完成する。ウォータフォールモデルの工程は、基本的にこの手法に準じたものである。

(類)トップダウンプログラミング

段階的統合化[stepwise integration]

段階的詳細化の技法で作られた成果物は、大きなシステムの部品に相当するモジュールであるが、これらを結合し、順次大きなシステムにしていく工程をいう。システムを構築するには、単体のモジュールを結合して次第に大きなモジュールにしながらテストとデバッグを繰り返すボトムアップテストと、段階的詳細化と同じ方向で上位のモジュールから順次結合しながらテストを繰り返すトップダウンテストがある。

探索[search]

主記憶、媒体に記憶してあるデータ(レコード)を、特定の項目(コード

タ

や名前)をキーとして探すこと。アプリケーションソフトの「検索」にほぼ同義。一般的な探索の技法には、探索範囲をしらみつぶしに探す技法、探索範囲を次第に狭めていき目的のデータに達する技法、データの所在を演算、またはテーブルを参照して求める技法などがある。

配列やテーブルなどのデータ探索の代表的な手法には、(1)探索する集合を走査しながら検査する線形探索法(逐次探索法)、(2)探索する集合を二つに分割しながら検査する二分探索法がある。このほか、ある種の演算によって探索するハッシュ法、テーブルを参照するテーブルルックアップの手法などもある。

単精度[single precision]

数量をどの程度まで識別できるように表すかの精度をいう。単精度では1個の数を表すとき、機械語の1語を用いる。(関)倍精度

単体テスト[unit test/module test] (同)モジュールテスト

タンデム結合システム[tandem system]

CPUを直列につなぎ、それぞれを処理状態に応じ使い分けたシステム。タンデムは3頭だて馬車の意味。主処理の前に前処理を行う装置をフロントエンドプロセッサ(FEP)、主処理後の後処理を行う装置をバックエンドプロセッサ(BEP)という。

ダンプ[dump]

ある時点でコンピュータ内部または記憶装置の内容を原形に近い形で出力すること。プログラムやデータのチェックに用いる。ダンプの結果をダンプリストという。次のような手法をよく用いる。

(1)デバッギングの際に、必要により、主記憶メモリの内容を出力することをメモリダンプ(memory dump)、ファイルの内容を出力することをファイルダンプ(file dump)という。

(2)プログラムのテストランの最中に、指定した時点で出力することをスナップショットダンプ(snapshot dump)といい、プログラムが異常終了した時点で出力することをポストモーテムダンプ(postmortem dump)という。

(3)チェックポイントダンプ(check point dump)は、プログラムの実行過程のところどころで記憶されている処理結果を取り出すダンプで、障害回復のときに利用する。

(4)静的ダンプ(static dump)はプログラムの実行中の特定時点や終了時点でダンプすることで、通常オペレータや監視プログラムの制御下で

行う。動的ダンプ(dynamic dump)はプログラムの実行中にダンプすることで、通常プログラムの制御下で行う。

チ

チェックコード[check code／check character]

転送、または伝送したデータの誤り検査のために送信側が付加するコード。冗長検査符号ともいう。データの入力があった場合、受けた側がデータに同様の処理をして、チェックコードと適合するか確かめる。符号の種類や長さや目的によりさまざまに称される。

(1)データが文字の場合

パリティチェックが一般的。付加するコードをチェックビットという。

(2)データが数値で、冗長符号が基数の1桁の場合

各桁に重みをつけて加算して、末尾の1桁、または基数で割った剰余、または下位何桁かをコードとする。付加するコードをチェックディジットという。各桁の重みが1(各桁の合計)の場合、特にチェックサムという。

(3)データがブロックである場合

伝送するブロックごとにブロック検査文字(block check character：BCC)を付加する。基本形データ伝送制御手順で使用する。複数の文字の場合、ブロック検査文字列(block check sequence：BCS)ともいう。

(4)データがフレームである場合

一定の演算で算出したコードを用いる。1ビットなら水平パリティチェックが一般的。2ビットなら群計数チェック、1語ならフレームチェックシーケンスという。

また、付加する冗長符号の長さにより、次のように呼ぶこともある。

- **1ビットの場合**：チェックビット(check bit)
- **数字1桁の場合**：チェックディジット(check digit)
- **文字1字の場合**：チェックキャラクタ、検査文字(check character)
- **文字列の場合**：チェックシーケンス、検査文字列(check sequence)

チェックシステム[check system]

データを処理するとき、出力した処理結果が、正確で信頼のおける内容

であるかをチェックするシステム。早期に、厳密に、効率的にチェックし、正しいデータに修正する方策を考慮する。データチェックは次の各段階に設ける。
(1)データ発生段階のチェック(入力原票のエラー)
(2)データ入力段階のチェック(入力操作のエラー)
(3)データ処理段階のチェック(データ項目やレコードの不備)
(4)結果の出力段階のチェック(出力装置との不整合、データ項目と使用権)

チェックポイント[check point]

トランザクション処理システムにおいて、プログラムの処理中に生じる障害に備え、処理の実行過程を一定の間隔で記録するように定めたポイント。

障害が発生してデータの内容を破壊した場合でも、直前のチェックポイントまでの保存データとチェックポイント以後のジャーナルファイルを用いて、必要なファイルを再現し実行を再開できる。この再実行をチェックポイントリスタートと呼ぶ。(参)ロールバック

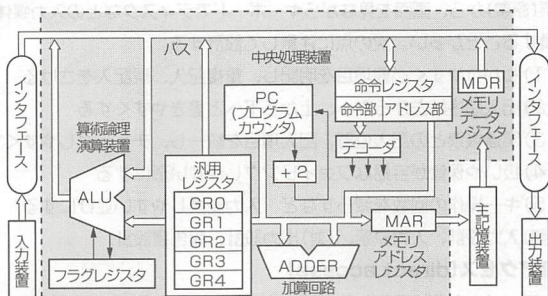
逐次再使用可能[serially reusable]

(参)再使用可能プログラム

逐次処理[serial processing] (同)順次処理

逐次探索[sequential search] (同)線形探索

中央処理装置[central processing unit：CPU]

コンピュータの装置の中で、ほかのすべての装置の動きを制御する制御装置と、四則演算や比較判断を行う算術論理演算装置をひとまとめにした装置。CPUと略称することが多い。中央処理装置には上記の装置とともに各種機能を有するレジスタ群を置き、内部バスラインで結んである。実際にはLSI上のひとつまたは複数個のモジュール内に形成してある。こうしたLSIをマイクロプロセッサ、またはMPU(microprocessor unit)とも呼ぶ。入力装置、出力装置、補助記憶装置などは中央処理装置に対比して周辺装置という。

中間言語[intermediate language／transaction code]

高級言語と機械語との間に設けた仮想のコンピュータのための機械語。中間コードともいう。高級言語からの翻訳の効率とハードウェアに対する効率の両方を考慮した結果生まれた。高級言語のプログラムはコンパイルの際に、いったん機種に依存しない中間コードに置き換えたあと、機械語(機種に依存)に翻訳する。また、各機種用に中間コードを解釈・実行するインタプリタを用意しておけば、同じ中間コードを異なる機種で実行可能である。Javaはこの方式を採用している。

チ

中間コード[transaction code] (同)中間言語

注釈[comment／annotation／remark／note]

ソースプログラム中にプログラマが覚え書きとして記述する文や行。コメントともいう。読みやすいソースプログラムを作るために、モジュールのはじめに標題等をつけるのに使う。注釈の書き方はプログラム言語によって異なる。この記述はコンパイルの際には無視し、プログラム実行時には何の影響も与えない。ワープロソフトにも印字しない注釈行を書く機能がある。

抽出[extraction]

検索の一機能で、指定した条件を満たすデータを検出すること。検索範囲、検索条件を指定し、新たな表を作ることができる。

抽象[abstraction] (同)データ抽象

チュートリアル[tutorial]

入門書、手引書。初めて使うソフトウェアの機能について、ユーザが大まかな操作を無理なく習得できるようにサンプルやアドバイスを織り交ぜながらやさしく解説を行う文書や図書。近年は、印刷物としてのマニュアルだけではなく、ビデオテープやパソコン上でデモンストレーションを実行したり、対話形式でステップアップ学習ができるプログラムなども多い。

帳票設計[document design／source document design]

入力設計段階での入力原票(原始帳票)の設計をいう。データは、原始帳票(原票)から、画面を見ながらキーボードでディスクなどの入力媒体に変換することが多い。次の点に注意して設計する。

(1)わかりやすく必要項目を明記し、重複記入、誤記入をさける
(2)記入順序は左から右へ、上から下へと書きやすくする
(3)関連帳票との記入位置、記入項目を統一し、チェックしやすくする
(4)扱いや保管が容易なファイリングしやすい形にする
(5)キー操作の回数を減らすなど、入力変換しやすい並びにする

(関)入力設計、入力原票、(対)出力設計、報告書設計、

直接アクセス[direct access]

ファイル中の特定のレコードに対し、その記録順序とは無関係に、また

前回アクセスしたデータとは無関係に指定レコードだけをアクセス、処理する方法。順次アクセスに対比される。オンライン処理でのデータ照会業務などを行うときには必須のアクセス法である。データの特定は、レコードのキーの値をもとに行う方法とデータの記憶アドレスを指定する方法がある。

直接アクセス記憶装置[direct access storage device：DASD]

データのアクセスタイムが、そのデータの記憶場所だけに依存し、直前にアクセスしたデータの記憶場所によらない記憶装置。磁気ディスク記憶装置など、ランダムプロセッシング(直接アクセス)が可能な記憶装置の総称。ダスド(DASD)ともいう。

直接アクセスファイル[direct access file]

直接編成、索引順編成のファイルで、直接アクセスが可能なファイルである。(対)順編成ファイル、直接アクセス記憶装置

直接アドレス指定[direct addressing]

命令語のアドレス部で、データの記憶場所を主記憶装置の物理アドレスで直接的に指定する方式。

(対)間接アドレス指定、(参)インデックスアドレス指定

直接アドレス方式[direct addressing]

直接編成ファイル作成方式のひとつ。キー項目の昇順に並べた入力レコードを、キー値と等しい番地(磁気ディスク装置内の格納アドレス)に、1件ずつ格納する記録方式。すなわちレコードのキー値をそのまま、シリンダ番号(CC)、ヘッド番号(HH)、レコード番号(R)に当てて直接的にアドレスを指定する。

実際にはレコードのキー値をそのまま使用すると無駄な格納場所がたくさんできてしまうので、キー値に簡単な計算をほどこして求めたアドレスや、キー値と格納番地の対比が記録してある参照テーブルから求めたアドレスに格納する間接アドレス方式をとる。

(参)間接アドレス方式、(参)ハッシング

直接編成ファイル[direct organization file]

レコード内のキーをもとに算出した直接アクセス記憶装置(DASD)上の番地をアクセスするファイル編成によるファイル。キー項目から計算で記憶場所を算出し、直接アクセスできるため、乱処理に適している。このため記憶はランダムになり、順処理は可能だが能率は低下する。ランダムファイルともいう。記憶は1ブロック1レコードとなる。各ブロックは、

チ

それぞれの位置などを表す情報(カウント部＝シリンダ番号、ヘッド番号、ブロック番号、キー長、データ長など)とデータから構成する。DASD上にのみ作成できる。

著作権[copyright]

著作物の著作者が、著作物を複製し、頒布することによって死後50年間を含め排他的に利益を受ける権利(1971年に施行)。1985年の著作権法の改正で、コンピュータのプログラムやデータベースも著作物として保護することになった。複製権、貸与権、翻訳権、翻案権など、利用形態の違いにより、多数の種類に分かれる。プログラム著作権は、ソースプログラムばかりでなくオブジェクトプログラムにまで及ぶ。しかし、プログラム作成時に使用した言語、規則やアルゴリズムまでには及ばない。

ツ

追跡プログラム[trace program／tracer]　(参)トレース

通信ソフトウェア[communication software]

コンピュータシステムが、他のコンピュータや端末とネットワークを介して通信するための機能を提供するソフトウェア。おもな機能は、プロトコルやモデムの設定機能、自動ダイヤル機能、自動ログイン機能、ダウンロード・アップロード機能、端末エミュレーション機能、コード変換機能など。

通信ネットワーク[communication network]

端末間を結ぶ機器(交換機や中継機など)と伝送路で構成するデータ伝送のための通信媒体、通信規約(プロトコル)の集合。通信網ともいう。それぞれの目的で作られた、さまざまな規模の通信ネットワークがあるが、システムとしては以下のような機能、品質に集約できる。

(1)大量の情報を高速に、効率よく伝送する

(2)ネットワーク内の任意の2点間で情報を交換する

(3)誤りのない信頼度の高い伝送を行う

また情報処理の分野では、オンラインシステムが構築可能な回線網をさす。通信ネットワークの機能を利用してコンピュータの本体とは離れた場所にある複数の端末装置を結んで情報資源を共有し、コンピュータを効率よく利用する。このようなシステムを特にコンピュータネットワークという。さらに通信ネットワークは次のような観点で分類できる。

(1)情報内容(音声・映像・画像・テキスト・ファイルなど)

(2)ネットワークの規模(国際・国内・構内・システム内など)

(3) 伝送方式(アナログ・ディジタルなど)

(4) 交換方式(回線交換・蓄積交換など)

(5) サービス内容(公衆網・専用線網・移動体通信など)

突合せ[match／matching]

複数のファイルやレコードについて、データや特定のキーを比較して、同じであるかどうかを照合すること。一般に突合せにより、必要データの抽出、編集、マスタファイルの更新などを行う。

ツリー構造[tree structure] (同)木構造

テ

底①[radix]

実数の基数記数法で用いる基準となる数。数をa×rnの形で表すとき、aを仮数、rを底、nを指数という。

底②[base]

基底のこと。プログラムの実効アドレスを計算するときの基準となるアドレスを特にベースアドレスという。

(関)ベースアドレス

ディジタル署名[digital signature]

情報の送信者が本人のみが知る秘密かぎにより情報を暗号化して送信すること。電子メールなどを使用したペーパレスシステムにおける承認印の代わりに用いる。受信者は公開復号かぎをもっており、それで復号できる。送信者が送ってきた電文を公開復号かぎで解読し、送信者の名前や日付が現われれば暗号化かぎをもった送信者本人からのものであることが証明できる。すなわち、ディジタル署名は暗号化できるのが限られた人のみで、復号は受取人なら誰でもできるという機能を実現している。(参)認証

低水準言語[low level language]

汎用プログラム言語を大きく「機械向き言語」と「問題向き言語」に分けたときの機械向き言語。具体的には、機械語やアセンブラ言語のことをさし、コンパイラ言語に対比して低水準言語という。(対)高水準言語

定数[constant／literal]

プログラム中で用いるデータの値と特性を明示する文字列(綴り)で、固定した値、あるいは変更しない値、データ項目。数値定数と文字定数がある。一般に引用はできるが、変数とは異なり、それに対する値の代入はできない。ALGOLにはこの用語はない。Fortranでは整定数、実定数といい、BASICでは、数値定数、文字列定数などという。COBOLでは、値定

数といい、数字定数、文字定数、表意定数に分類される。

ディスクアレイ[disk array] （同）RAID

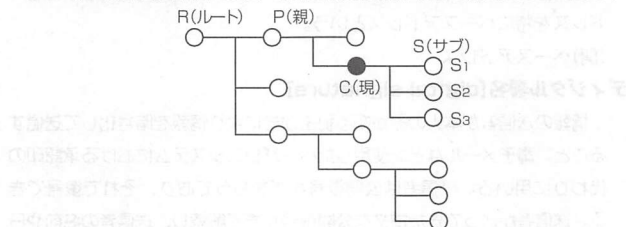
ディレクトリ[directory]

区分編成ファイルでは、順編成ファイルをメンバというサブファイルに分割し、各メンバを管理している記録をディレクトリ（登録簿）という。登録簿には各メンバのファイル名（プログラム名）や大きさ、作成日、記憶領域の先頭アドレスなどを書き込んである。似たものにインデックス（索引順次編成ファイルでキーの値で作成する索引）やカタログ（磁気記憶媒体上に作るファイルを管理するために作成する）がある。

テ

ディレクトリ管理[directory management]

ファイルの数が多い場合、一媒体にすべてのファイルを並列に保存すると煩雑になるため、その内容によりグループ化し体系的に管理すること。

UNIX系OSやMS-DOSでは階層的なファイルシステムを利用して、ひとつの媒体のルートディレクトリの下に分岐ディレクトリ（サブディレクトリ）を作り、その中にファイルを整理して保存する。さらに、その下に分岐ディレクトリを作ることもできる。これらのディレクトリをうまく構成すると、ファイルの管理が容易になり、利用の際の混乱や誤りを防ぐこともできる。

あるディレクトリを操作するとき、そのディレクトリをカレントディレクトリ（図ではCとする）、そのすぐ下位のディレクトリをサブディレクトリ（図のS）、そのすぐ上位のディレクトリを親ディレクトリ（図のP）という。最上位のディレクトリをルートディレクトリ（図のR）、ユーザにとっての最上位ディレクトリをホームディレクトリという。GUI環境のOSでは、同様の思想を「フォルダ」で実現している。

データ[data]

文字・記号・図形・図表・音声・信号・身ぶりなどのように意味を付与した表現。伝達・判断・処理の対象のこと。

①人間またはコンピュータが処理する数・文字・記号の集まり。コンピュータで処理するデータはその内容の表現に適した型をもち、値を有する。まだ処理していない単なる諸事実・概念・指令を、情報と区別してデータということもある。（参）情報

②データ処理の単位。レコードという。レコードの中のそれぞれの内容を項目、関連したレコードの集まりをファイルという。

(参)項目、レコード、ファイル

データ圧縮[data compression／data compaction]

データの意味を変えずに、あるいは再現できる範囲内で、データサイズ(容量)を小さくする変換。単に圧縮ともいう。データ圧縮ユーティリティ(data compression utility)は、もとの情報を損なわないように、ファイルのデータ量を削減するプログラム。繰り返し出現する冗長データや不必要な文字に着目して、符号変換を行い、使用時に展開(解凍)する。

データ圧縮は記憶媒体の効率的使用、伝送時間の節約等にねらいがある。通信回線による大量のデータ伝送では、圧縮や伸張にかかる手間や時間を加えても、データ伝送時間が短くなり通信コストが少なくてすむ。

(1)文字データやプログラムファイル：一般的に空白や同一パターンの繰り返しが多いデータほど圧縮率が高い。パソコンでも日常的な手法で、利用者がまったく処理を意識しなくても入出力時に自動的に圧縮、伸張を行うソフトウェアもある。

(2)音声データや画像データ：再現に差し支えない範囲内で圧縮する。テキストデータに比べて格段に容量が大きくなるため、保存、伝送の際に圧縮処理は欠かせない。標準的な画像データの圧縮手法として静止画用のJPEG、動画用のMPEGがある。他にも画質が若干劣化し、伸張してもデータが完全復元できないが、圧縮率が特段に高いものもある。

データウェアハウス[dataware house：DWH]

企業内のさまざまなデータを保有し、企業戦略の立案や意思決定を支援するデータを提供するシステム。またはその目的のためデータベース構築の思想。ソースデータを蓄積するデータベースやデータを抽出するソフトウェアの普及、エンドユーザコンピューティングの充実によって可能になった。例えば、流通業では、POSで収集したデータを単に仕入れ・販売業務に供するだけでなく、別の部の担当者が目的に応じて、時間帯と客筋、客筋と売れ筋商品の関係を明らかにし、販売戦略を立案できるような分析を行う。

データ型[data type] (同)型

データ管理[data management]

オペレーティングシステムの主要な機能のひとつ。処理プログラムが入出力命令を出すと入出力装置にアクセスする。その際に物理的な入出力装置の特性を意識しなくてもよいように論理的なアクセス手段を提供する。

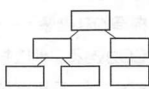
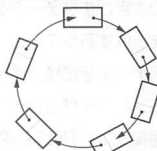
オペレーティングシステムのデータ管理は、ファイルを中心としたコンピュータシステム内のデータを統一的に管理することが目的である。その働きには、ブロック単位やレコード単位でのデータへのアクセス以外に

も、データの編成方法とそのアクセス法、カタログ、位置指定、記憶、検索、保守、データの保護などがある。

データ記述言語[data description language：DDL]

データベースの定義に関連する命令群で、DBMSが規定しているスキーマ定義言語。NDLとSQLが規定する用語である。主として以下のような機能をもつか、ユーティリティプログラムでサポートしている。(1)スキーマの生成機能、(2)実体ファイル(レコードファイルやインデックスファイル)の作成、(3)実体ファイルのカタログ作成、(4)既存データベースの情報提供

データ構造[data structure]

データ要素がどのように関連しているかという論理的構造。データモデルともいう。情報の実体を表現できる構造、アクセス側から制御できる構造、ファイルを構成できる構造などに関連する。線形構造、木構造、網構造、線形リスト構造、環(リング)構造等のモデルがある。プログラム設計に際しては、データの内容、処理方法、記憶方法などを考慮して決める。代表的なデータ構造については、探索、追加、削除などのアルゴリズムの研究が完成している。

データ項目[information for data items]

①データベースのデータ定義情報の一部で、ファイルのレコード内のデータ項目の名称や属性、長さ、件数、内容説明まで含む。

②COBOLでは、「項目」に同義。(同)項目

データスペース[data space]

VSAMデータセット用にDASD上に割り当てられた記憶域。VSAMデータセットは必ずデータスペースの中に作成される。また、ひとつのデータスペースには、複数のVSAMデータセットを作成することができる。

データスペースに作られたVSAMデータセットは、いくつかの制御域(control area：CA)に分けて管理される。この制御域は複数の制御インタバル(control interval：CI)によって構成され、この中に複数の論理レコードが記憶される。(参)制御エリア、制御インタバル

データセット[data set]

ある目的のために、関連するデータ項目を一定の並び(入力順またはキー順)で編集、記述したデータの集まり。データを記憶域に格納し、あるいは記憶域から取り出すときに、データ管理が一括して取り扱うデータの単位となる。ファイルと同義で扱われることが多いが、VSAMではデータセットを用いる。

(例)キー順データセット、入力順データセット、相対レコードデータセット

データ操作言語[data manipulation language：DML]

DBMSのもとで、データベースのデータを読み書きするための言語。親言語(COBOLやFortranなどの汎用プログラム言語)でアプリケーションプログラムを開発する場合(親言語方式)や利用者が直接的にデータを照会する場合(独立言語方式)に使用する。

データ抽象[data abstraction]

コンピュータ処理の対象となる実世界のデータを分析し、処理に適したデータ構造を見い出し、グループ化し、データ間の関係を明らかにすること。実世界のデータはさまざまであるが、コンピュータ処理を行うためには、対象となる実体を表現するデータを抽出し、そのデータの属性、構造、性質を考えて処理のアルゴリズムを考えなくてはならない。コンピュータ処理では、基本的なデータ構造についてはデータ処理の方法が確立しているので、処理の対象となるデータ構造と処理のアルゴリズムがわかれば、その部分については基本操作を適用することができる。このような観点からは、プログラミングは、処理対象のデータを、コンピュータシステムで処理可能なデータ型に対応づけることに相当する。

データ定義言語[data definition language：DDL]

データベースの定義に関連する命令群で、DBMSが規定しているスキーマ定義言語。NDLとSQLが規定する呼称で、CODASYLではデータ記述言語(data description language：DDL)と呼んでいる。

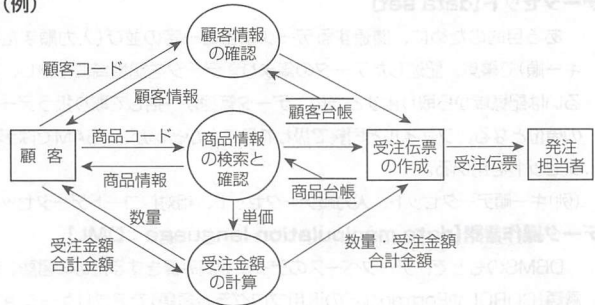
(同)データ記述言語

データディレクトリ[data directory]

データベースを定義するスキーマを構成する表、ビュー、権限の定義情報を登録する辞書。カタログ、またはデータディクショナリともいう。データディレクトリへの定義情報の登録はDBMSが自動的に行う。ユーザはその内容を、SELECT文で問い合わせることができる。

データフローダイアグラム[data flow diagram：DFD]

システム要件の定義を行う構造化分析に使用する図表のひとつ。基本的にはシステム間(モジュール間)のデータの流れに注目し、システムとデータの関係を図的にモデル化する技法。この段階では、データの物理構造

(例)

やハードウェアについては極力触れない。システムの記述を行う場合、次の4つの基本構成要素を使用する。

(1)データの入力元(源泉・発生源・情報源)／出力先(吸収・行き先)：名前を四角形で囲む

(2)処理(プロセス)の内容(処理の機能)：名前をバブル(円形)で囲む

(3)データフロー(データ名とデータの移動)：名前つき矢印で表す

(4)データストア(データ保管物・ファイル)：名前を2本の直線ではさむ

データベース[database：DB]

相互に関連のあるデータを重複しないように集め、その内容を構造化し、多目的な利用(検索や更新)が効率的にできるようにしたデータの集まり。プログラムによる利用もできるが、プログラムから独立しても利用できる点でファイルとは異なる。昨今は、小型コンピュータの普及、個人ユーザの増加、マルチメディア化の要求の中で、従来のシステムの拡張や簡易化が進み、音声・画像などを含むデータベースシステムも開発され、多様化している。

【データベースとファイルの違い】

データベース	ファイル
・多くの利用者や複数の適用業務プログラムで共用されることを前提にしたデータの集合単位 ・データの構造や属性が変わっても，適用業務プログラムの変更が少なく，安定している ・データはシステム全体で，重複なく効率よく統合管理され，データの関連データベースシステムが管理する。	・ファイルの定義や保守は適用業務プログラムに従属している。 ・異なるファイル間にデータの重複が多く，関連は利用者の判断に委ねられはる。 ・特定の利用目的，適用業務に従属する局所的なデータの集合単位。

【データベースの分類】

また、データベースには、利用者の範囲から次のような種類がある。

(1) 企業や機関が作る独自のデータベース、利用者が作る個人のデータベース

(2) 情報サービス業者がデータを提供する商用データベース

後者はデータ通信とタイムシェアリングシステム(TSS)を利用して、同時に多数の端末から共同利用できる。

また、データベース構築、管理の際には、処理方式や共有するデータの特性とデータ間の関係に注目して構造をもたせるが、この構造、管理の観点から次のように分類できる。

(1) レコード間に階層関係をもたせた階層型データベース

(2) レコード間につながりをもたせたネットワーク型データベース

(3) レコードの集合間に対応関係をもたせた関係データベース

(1)、(2)を構造型データベース、(3)を非構造型データベースという。

【データベースの歴史】

データベースの考え方は、1960年代から議論されていたが、1963年、米GE社のネットワーク型データベースIDS(Integrated Data Store)、1987年、米IBM社の階層型データベースIMS(Information Management System)により実現した。

1967年、CODASYL内にデータベースを研究するDBTG(Data Base Task Group)が組織された。DBTGの提案やレポートをきっかけにデータベース言語の標準化の動きが生まれ、CODASYL型データベースと呼ばれた。一方、別のデータベースの考え方もあり、1970年、IBM社のE.F.Coddが関係データベースを提案、1980年代になって米IBM社のSQLやOracle社のORACLEなど続々と製品が開発された。これも規格化の動きの中で、SQLが1987年にISO規格になり、JIS規格となった。

データベース管理システム[database management system：DBMS]

データベースとその利用者の間に介在してデータベースの管理を専門に効率よく行うためのソフトウェアシステム。この管理システムにはデータベース記述言語(DDL)やデータベース操作言語(DML)を用意してある。

データベース管理者はこれを用いてスキーマやサブスキーマの定義、データの検索や追加、修正、削除などが可能になる。

【データベース管理システムの役割】

(1) 利用者が定義するデータの論理的構造について、記憶装置上のデータの物理的構造との対応をとる。

テ

(2) 利用者がデータベースからデータを取り出し、操作するプログラムによって、データベースの構造が変更されないようにする。
(3) データを定義するメタデータを、プログラムから独立に定義したデータ辞書(DD/D)で管理する。
(4) データの正常性、整合性が失われないように制御を行う。
(5) 好ましくないアクセス、不正なアクセスからデータを保護する。
(6) データベースの利用者にデータベース操作言語を提供する(独立言語方式と親言語方式がある)。
(7) データベース運用のためのユーティリティソフトウェアを提供する。
(関) データベース、スキーマ

テ

データベース言語[database language]

データベースシステムを利用するための言語。次の3段階の言語機能をもつ。
(1) データを定義する段階。
(2) データを読み書きする段階。
(3) 得たデータをもとに作表・演算・グラフ表示を行う段階。
この全体をひとつの言語処理系でまかなう方式を独立言語方式、(1)の段階をデータベース言語がになって、(2)、(3)の段階を汎用プログラム言語で行う方式を親言語方式という。

データベースサーバ[database server]

クライアントサーバシステムの中で、データベース機能を提供するサーバ。ファイルサーバの一例であるが、特に高速な検索要求に答えるため、キャッシュ機能をもつ。ソフトウェアの例にはアメリカのインフォミックス社のInformix、オラクル社のORACLEなどがある。

データベース再編成[reorganization of database]

データベースの効率的利用のため、内部スキーマを変更することなく、データの物理的実現方法を変更すること。例えば、アクセス頻度の高いデータを高速アクセス媒体上に移したり、検索経路が短縮するように並べ替えたり、冗長データを削除することなど。

データベース操作[database manipulation]

データベースのデータを創成(creation)、再編成(reorganization)、再構成(reconstruction)を行い、読み書き(検索、更新)すること。そのための機能をもつ言語をデータベース操作言語という。

データベースソフト[database software]

アプリケーションパッケージソフトの一種で、簡易なデータベース機能を有するソフトウェア。表計算ソフトや帳票作成ソフト、グラフ出力ソフト、ワープロソフトなどと組み合わせて利用することができる。部門内や

個人で利用する小規模なデータベースの構築に適している。

データ変換[data conversion]　(同)コード変換、型変換

データモデル[data model]

データの意味とデータ間の関係を認識、記述するための概念的な構造とそれを操作するための体系。データベースで扱うデータモデルは、

(1)人間が認識する概念的な段階のもの

(2)プログラムで処理できる論理的な段階のもの

(3)記憶装置の構造や格納する効率を考えた物理的な段階のもの

で異なる。また、概念的な段階のデータモデルでも、(1)階層型モデル、(2)網モデル、(3)関係モデル、(4)オブジェクト指向モデルなどがあり、それぞれのデータ構造に適したデータベースシステムが実現されている。

テーブル[table]

①互いに関係のある項目が、その相対的位置によって識別できるようにデータを配列したもの。その各項目は、ひとつ以上の引数(添字)またはキーであいまいなく識別できる。表ともいう。引数に異なる数を当て異なるデータを示すことができ、本来なら複数の式が必要な場合でも、引数の値を変えることによりひとつの基本式で処理可能となる。コード変換テーブルなどがこれに相当し、データ処理で多用する。(類)配列

②表計算ソフトや関係データベースソフトで、互いに関係のあるデータを縦横形式の表組になるように配列した表。その各項目は、数式などを用いてひとつ以上の引数(添字)であいまいなく識別できる。単に表ともいう。広く表計算ソフトやワープロソフトで、縦横形式の表組で出力した表をも指す。これらは表組ユーティリティを使って定義できる。

テーブルサーチ[table search]

テーブルは定めた規則により並べた同質の特性をもったデータの集まり。一般に配列と同じ。テーブルサーチとは、特定の項目(キー)を指示することによりこのテーブルの中から、条件にかなったデータを探すこと。

テキストエディタ[text editor]　(参)エディタ

テキストファイル[text file]

テキスト形式のデータを内容とするファイル。例えば、(1)テキストエディタで編集したソースプログラム、(2)情報交換用のCSV形式のテキストレコード、(3)ワープロ文書の本文データなど。多くのエディタやワープロソフトで読めるため、情報の交換用に使用する。ただし、書式がついていないので、ワープロソフトで設定した書体や文字サイズは反映されない一様な文字レコードである。

デクリメント[decrement]

減分。ある値から一定の値を繰り返し減算していく場合の、減算する単位値のこと。例えば、プログラムで、ループを1回実行するたび制御変数やポインタから一定の値を減算する場合、減算していく値が減分である。

FortranのDO文、COBOLのPERFORM文、BASICのFOR…NEXT文などでは、この減分をパラメタとして指定できる。なお、同様に加算する場合には増分という。減分は負の増分ともいえる。

(対)インクリメント

デシジョンテーブル[decision table]

問題を分析する場合に考慮すべきすべての条件と、条件に対しとるべき処理を列挙した表。決定表ともいう。問題の解を見いだすための説明やプログラム設計の際の補助的な図式として使用する。

		条件記入欄								
条件表題欄	英数合格	Y	N	Y	Y	N	N	Y	N	Y=Yes(成立)
	情報処理合格	Y	Y	Y	N	Y	N	N	N	N=No(不成立)
	プログラミング合格	Y	Y	N	Y	N	Y	N	N	
行動表題欄	進　　級	×								×=とるべき行動
	進級不可					×	×	×	×	
	レポート提出		×	×	×					
		行動記入欄								

テスト[test]

開発したコンピュータシステムの機能が設計仕様と一致するか確かめること。作成したプログラムの妥当性、システムや装置の正当性の検査。誤りを検出する目的でプログラムを試し実行する。試験、検査ともいう。

テスト技法は開発技法により変わり、その目的や手法によりさまざまな呼び方がある。

(1)テストはシステム開発の各段階で計画的に繰返し行う。

①**設計段階**：(関)机上デバッグ、ウォークスルー

②**プログラム開発段階**：(関)トップダウンテスト、ボトムアップテスト

③**完成段階**：(関)総合テスト、受入れテスト

(2)モジュール分割したプログラムでは、次のテストを行う。

(関)モジュールテスト、結合テスト、システムテスト、総合テスト

(3)モジュールの結合順序の違いで、次のテストがある。

(関)ボトムアップテスト、トップダウンテスト

(4)テストケースの設計方法の違いで次のような呼び方がある。

(関)ブラックボックステスト、ホワイトボックステスト

(5)テストの目的により次のような呼び方がある。

(関)引渡しテスト(受入れテスト)、並行テスト、負荷テスト

テスト計画[test planning]

テストを行うにあたって立案する計画。テストの対象となるシステムやテストの段階で異なり、通常、開発部門におけるシステムテストとエンドユーザ部門の運用テストの2段階で行う。

開発部門のシステムテストでは次のテストを行う。

(関)機能テスト、性能テスト、障害回復テスト、負荷テスト

エンドユーザ部門においては次の観点から計画し、テスト仕様書にする。

(1)関連部門との整合性(他システムとのデータ、ファイルの互換性)
(2)オペレータの操作性(伝票・帳票・入力方法など使いやすさ)
(3)並行テストの際は、新旧システムの処理の区分(業務の混乱を避ける)
(4)テスト期間、トラブル時の修正期間、再テスト項目
(5)新システムに切り換え、移行する時点の判断基準

テストケース[test case]

開発したシステムの効率的かつ効果的なテストを行うための必要十分なテスト条件。テスト条件が場合分けされたのが、テストケースであり、テストケースの設計はテスト計画の段階における重要な作業になる。また、プログラム中のすべてのルーチンをもれなくテストできるようなデータを用意するほか、本来なら実データにないはずの例外的なデータについてもチェックし、システムの信頼性を確保する必要がある。

(関)ブラックボックステスト、ホワイトボックステスト

テスト支援ツール[program testing tool]

プログラムのテスト条件や環境を整備するプログラム群の総称。これらのツールを使用してテストすることをキャット(CAT：computer aided testing)という。

さまざまな種類のツールがあるが、大きくふたつに分けられる。

(1)プログラムの記述を解析するツールや、実行しながら解析するデバッギングエイドなど
(2)テストの実施を支援するドライバやスタブのシミュレータ、テストデータの作成プログラム(test data generator)など

テストデータ[test data]

プログラムを検査するために準備するデータ。入力としてありそうなデータからめったにないデータまで組み合わせて、プログラムの全ルーチンを網羅するように適当な数だけ準備する。期待する処理結果も準備しておき、実行させてみた結果と照合する。テストの目的により、データ量、分布や頻度、境界値や例外値、エラーデータ、ランダム性などを

テ

考慮して作成する。

テストプログラム[test program]

コンピュータのハードウェア、ソフトウェアが正常に動作するか否かを検査するためのプログラム。テストのために、特別のルーチンを付加したり、チェックポイントを設けて記憶内容をダンプさせる。

手続き型言語[procedure oriented language]

処理の手続きを明確、容易に記述できるような体系をもったプログラム言語。プログラムは処理を指定する文の集合からなり、ハードウェアも原則として記述された文を記述された順序で実行する。命令もデータも同じ記憶装置内にあり、命令の実行により記憶内容を変更することで処理を段階的(逐次的)に進める。FortranやCOBOLをはじめ大部分の高水準言語がこれに相当する。(対)非手続き型言語

デッドロック[dead lock]

資源を共有している二つのプロセス(またはタスク)が、互いに相手の資源の解放を待って競合している状態、または、結果的に動きがとれなくなり、どちらもアクセスできない状態。例えば、共有不可の資源X、Yがあり、プロセスAがXを占有し、Yの解放を待っている。そして、プロセスBがYを占有し、Xの解放を待っている。排他制御により、どちらのタスクも新たな資源を占有することができず、デッドロックとなる。共有不可の資源としては、共有メモリやファイル、通信回線などがある。デッドロックは、資源利用の排他制御の副作用として発生する厄介な問題であり、システム全体の停止を引き起こすこともある。(参)排他制御

デバッガ[debugger] (同)デバッグツール

デバッギングエイド[debugging aid] (同)デバッグツール

デバッグ[debug／debugging]

プログラム中の誤り(形式エラー、文法エラー、論理エラー、設計エラーなど)を発見し、取り除き、修正する作業。誤りを俗にバグ(虫)という。そこでこの作業を虫取り(デバッギング)ともいう。プログラマがテストの前段階で行う。机上でプログラムを見直して誤りを発見することもデバッグである。文法エラーについては、コンピュータにプログラムを与えて翻訳させ、エラーメッセージを参照して行うのが普通である。論理エラーについては、テストデータを与え、出力をチェックして発見する。

デバッグツール[debug tool／debugging tool]

デバッグを支援するサービスプログラム。デバッガ、デバッギングエイドともいう。トレースプログラム(トレーサ)、ダンププログラム(ダンパ)などを総称する語。広義には、テスト支援ツール(例えば、テストデータ

ジェネレータ)、診断プログラム、シミュレータ(障害シミュレータ、装置シミュレータ)なども含む。バッチ処理で行うものと対話型で行うものとがある。

デファクトスタンダード[defacto standard]

業界標準。ISO、ITU-T、JISなどの標準化機構が正式に標準として制定または勧告をしていないが、多くの企業がある種の方式に合わせているため、それが事実上の標準となっているものをいう。

具体例には、TCP/IPプロトコル、セントロニクスインタフェース、シフトJISコードなどがある。デファクトスタンダードは、後に正式の標準となることもある。例えば、IBMのSDLCがHDLCに、ヒューレットパッカード社のHPIBがGPIB(IEEE-488)に、イーサネットLANがIEEE802.3にと、業界標準が正式の標準となった。

デフォルト[default]

①プログラム中に明示的に指定されていない場合にシステムが選ぶ、変数の属性や値。オペレータがコマンドの入力を省略したり、プログラムに変数の型などが明示的に宣言されていない場合、システムがあらかじめ決めてある値(デフォルト値)や型が指定されたと見なして処理を行う。

(参)暗黙の型宣言

②新規のハード、ソフトについて、メーカが多数のユーザの使用を想定して出荷時の製品に設定した標準的なパラメタ。ユーザがこれを変更して自システムに合わせることをシステム設定、環境設定、初期設定、カスタマイズなどという。

デブロッキング[deblocking]

物理レコードを複数の論理レコードに分解すること。外部記憶装置のデータの読み書きの効率を高めるため、レコードの読出し命令があると外部記憶装置はブロック(物理レコード)のままバッファに送り出す。このバッファ上のブロックを分解して、利用者、プログラムの処理単位(論理レコード)に変換する作業である。(対)ブロッキング

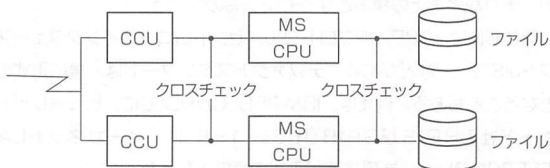
デュアル構成システム[dual system]

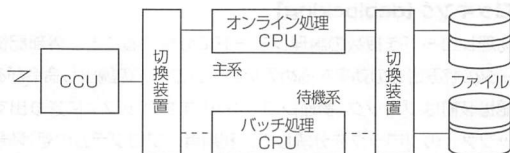
システムを二重に用意し、常に両方とも同時に動作し同一の内容となるようにしておくこと。一方が故障しても他方が正常であれば、全体システムに影響を与えない。高信頼性を要求するシステムを構築するときに使用する構成である。構成要素を可能な限りデュアル構成にし、筐体のなかに格納した高信頼性コンピュータにフォールトトレラントコンピュータがある。デュアル構成が完全にダウンする確率は一方の故障確率の2乗となる。したがって、デュアル構成の稼働率は次の計算式で表せる。

デュアル構成の稼働率＝1－{(MTTR)/(MTBF＋MTTR)}^{2}

例えば、一方の計算機が故障する確率が理論上0.001だとすると、デュアル構成全体がダウンする確率は0.00001となる。

また、障害時には故障システムを切り離すことにより、もう一方のシステムで処理を続けることができ、その信頼性は非常に高い。ホットスタンバイ方式ともいう。ただし運用コストは高価になる。

(参)マルチプロセッサシステム、デュプレックス構成システム

デュプレックス構成システム[duplex system]

2組のシステムを用意し、主目的の処理を行うシステム(主系)の障害発生時に、別のシステム(待機系)に切り替えるシステム構成。二重方式、コールドスタンバイ方式ともいう。オンラインシステムでは、信頼性の向上がかなり厳しく要求され、デュプレックス構成にしてシステムに障害が発生しても停止することなく稼働し続けるようにしたシステムが多い。デュアル構成に比べて、待機系では待機中に別の処理を行えるため効率的なシステム構成である。しかし、主系ダウン時の切り替えに時間がかかる、切り替え部分の障害や切り替え直後に新主系がダウンすると両系ダウンとなるなど、信頼性はデュアル構成に劣る。

テラ[tera]

テラは、10^{12}(＝1兆)を意味する補助単位。1テラバイト＝1000ギガバイト＝1兆バイト。今後予想される記憶装置やデータ処理量の大容量化を象徴する用語。

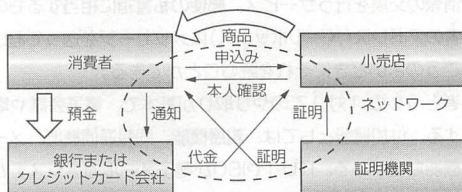
デリミタ[delimiter]　(同)区切り文字

電子計算機システム安全対策基準

(同)情報システム安全対策基準

電子決済[electronic account]

通信ネットワークを介した商取引きで、取引き相手を本人確認し、金銭の決済を行うこと。電子商取引(EC)の基本ではあるが、安全性を考えて

慎重な対応が必要との意見もある。利用者は、銀行やクレジットカード会社などの預金から小売店に代金決済する。第三者である認証機関が証明する場合に限り決済が可能になる方式も検討中である。(関)証明機関

電子商取引[electronic commerce：EC]

通信ネットワークを介した商取引きのこと。電子メール、ディジタル署名および電子決済により、ネットワーク上で取引きのすべてを済ませることも可能である。(類)ディジタル署名、電子決済、(参)電子データ交換

電子署名[digital signature]　(同)ディジタル署名

電子データ交換[Electronic Data Interchange：EDI]

取引き関係にある複数の企業間でコンピュータと通信回線を結んで定形業務のデータをやりとりすること。従来のビジネスプロトコル(業界プロトコル)も狭義のEDIといえる。

ISOによるEDIFACT(Electronic Data Interchange For Administration Commerce and Transport：行政、商業および運輸のための電子データ交換)の標準化の動き、国内では通産省がCIIシンタックスルールの制定、電信電話技術委員会(TTC)によるISDNとMHSの良さを取り入れたEDI-MHSの標準化など、EDIは業界の壁をこえた標準化という意味でその重要性が認識されるようになってきた。それを受け、1992年9月に国内の39の業界団体が集合して「EDI推進協議会」を設立した。(類)電子商取引

電子マネー[electronic money]

本来の意味は電子的な現金。電子技術により金銭の取引きを実現するシステム。ディジタルキャッシュ、電子通貨ともいう。現金で換金したもの、銀行やクレジット会社が発行したものが使えるようになりつつある。

(1) イギリスのモンデックスUK社が開発したモンデックスがある。これはICカードを現金の代わりに利用でき、専用端末でICカードに記録されている金銭の出し入れができる。

(2) インターネット上のオンラインショッピングには、プリペイド方式のウェブマネーが使えるが、残額はカードに記憶するのではなくカード発行会社のコンピュータが記憶する。

(関)電子決済

電子メールサービス[electronic mail service]

VAN、LAN、パソコン通信などの蓄積交換サービスの一種。システム内に利用者ごとにメールボックスを用意し、メールボックスを介して利用者間で情報の交換を行うサービス。郵便の私書箱に相当するもの。発信者が受信者のアドレス(メールボックス)にメッセージを送っておけば、受信者はつごうのよいときにそれを読むことができる。

利用者どうしの1対1でのやり取りが基本で、電子会議や電子掲示板と区別する。付加機能としては、親展機能、同報通信機能、メールボックス検索機能などがある。ITU-TやISOが電子メール通信システムの国際標準をMHS／MOTISという名称で勧告し、それに準拠した製品が開発されている。また、マルチメディアや各国言語に対応したインターネットのMIME規格に準拠した電子メールの普及が著しい。

転置式暗号[transposition cipher] (関)慣用暗号系

テンポラリファイル[temporary file]

①ユーザが、プログラムやデータを一時的に保存しておくファイル。

②OSが、あるジョブ内で必要に応じて一時的に作るファイル。ジョブの終了時に自動的に消去する。

ト

動画像符号化[motion picture encode] (関)MPEG

動画データ[animation data]

映画、ビデオ、アニメなどの動画をディジタル化したデータ。動画をディジタル化するとデータ量が膨大になる。そのため、処理装置の高速化だけでなく、データ圧縮、高速、大容量の記憶装置、高速伝送などの技術が重要となる。圧縮技術ではMPEG、記憶装置ではディジタルビデオディスク(DVD)、伝送技術ではATMが今後の標準となる可能性がある。

動的アクセス[dynamic access]

直接アクセスと順次アクセスを組み合わせてアクセスする方法。直接アクセスで特定のレコードをアクセスし、それ以降のレコードを順次アクセスにより処理すること。

動的ダンプ[dynamic dump] (関)ダンプ

動的呼出し法[dynamic access]

相対及び索引ファイルにのみ認められている呼出し法。ひとつのプログラムの中で、順呼出しと乱呼出しの両方を使用できる。

登録簿[directory]

区分編成ファイルでは、順編成ファイルをメンバというサブファイルに分割し、各メンバを管理している。その記録を登録簿という。登録簿には各メンバの名称(プログラム名、ファイル名など)や先頭のアドレスなどを書き込んである。似たものにインデックス(索引順次編成ファイルでキーの値で索引を作成する)やカタログ(磁気記憶媒体上に作るファイルを管理するために作成する)がある。

ドキュメンテーション[documentation]

文書(ドキュメントやマニュアル)の管理、あるいは与えた主題に関する文書自身の集合。書類または書類中に記録してある情報を作成、収集、編成、保存、引用、配布すること。情報処理の分野では、特にシステム開発に際して、計画書・設計書・仕様書・報告書・運用指示書・保守記録などに関する文書の作成・配布・保存をさす。開発工程のおのおのの段階で必要である。一般に文書化という。

ユーザ部門では、システム開発側が用意する運用指示書や操作説明書がエンドユーザ向けでない場合(難解な場合)、日常の運用のために簡易なドキュメントを作成、配布することも必要である。

特殊文字[special character]

文字集合の中で英字、数字、漢字、仮名、間隔文字のいずれでもない図形文字と制御文字。例えば、○+−¥などの記号文字、伝送制御文字など。ソフトウェアの操作、伝送データの中で特殊な役割をもった文字として使用する。例として、演算子、引用符、ワイルドカード、区切り文字、小数点、伝送制御文字

特殊問題向き言語[special purpose(oriented]language]

比較的広い分野のプログラムに使用する汎用プログラム言語に対して、特定の応用分野の問題を処理するために開発した言語。例えばCOGOやSTRESSなどの土木建設分野の言語、数値制御用のAPTなどがある。シミュレーション用に開発したGPSSは離散系に向いており、DYNAMOは連続系に向いている。プロセス制御用言語、数式処理用言語、グラフィックス用言語、記号処理用言語、論理設計用言語、データベース用言語などがある。(参)プログラム言語

独立あふれ域[independent overflow area]

(関)索引順次編成

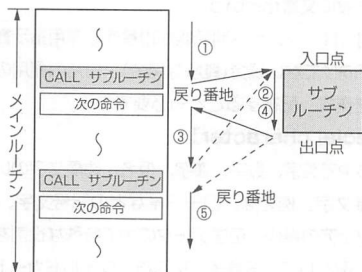
独立言語方式[self-contained system]

データベース操作言語(DML)の形態のひとつ。既存のプログラム言語とは独立して、そのDMLだけでデータベースの構築・操作・利用に必要な処理ができる方式である。利用者は親言語を使用することなく、DBMSに備

わっているDMLを用いて対話的にデータを操作できる。この方式は一般の利用者(エンドユーザ)が利用しやすいので、エンドユーザ言語(end user language：EUL)とも言われる。

JIS SQL仕様のものを会話式に利用できるようにしたものを提供しているメーカも多い。初期のデータベース言語に多かった方式で、従来の汎用プログラム言語にとらわれない新しい機能をもつ半面、これだけでは、既開発の汎用プログラム言語のプログラムとの間のインタフェースをとれないという欠点もあった。この欠点を補う方式が親言語方式である。現在は、双方の方式で利用できるデータベースシステムが多い。(対)親言語方式

閉じたサブルーチン[closed subroutine]

同一処理内容のルーチンをプログラム中に多数回組み込むことはせず、プログラム内に1個だけ挿入しておき、必要な箇所からそのつど呼び出して利用できるサブルーチン。(対)開いたサブルーチン

特権命令[privileged instruction]

スーパバイザモードだけで使用できる命令。コンピュータの状態の変更命令、主記憶保護キーの変更命令、入出力機器の起動や制御の入出力命令などがある。ハードウェア資源の管理・マルチプログラミングの制御は、監視プログラム(スーパバイザ)が行っている。複数のプログラムやユーザが資源を共有する場合、ある種の命令は誤って使用するとシステムを破壊する可能性がある。そのため、一般のプログラムでは使用を禁じ、特別なモード(スーパバイザモード)でのみ使用ができるようにした。(参)SVC

トップダウンテスト[topdown test]

階層構造化したモジュールの、最上位レベルからテストを始め、順次下位のモジュールを上位モジュールと結合してモジュール間のインタフェースをテストする方法。上位モジュールをテストするとき、その下位レベルのモジュールが未完成の場合、スタブというテスト用プログラムを使用する。

トップダウンテストでは、プログラムの中心となる部分(上位モジュール)が下位モジュールとの結合テストにより、繰り返しテストできる利点

がある。下位モジュールが未完成のテストの初期段階では、スタブを用いないとプログラム開発と並行してテストをできないという欠点もある。
(関)スタブ、(対)ボトムアップテスト

トップダウンプログラミング[topdown programming]

ソフトウェアの機能をモジュールに分割して、階層構造を明確にしたうえで、上位モジュールから下位モジュールへと段階的に詳細なプログラムを作成する技法。トップダウンアプローチ(topdown approach)ともいう。段階的詳細化技法ということもある。プログラムの全体像を意識したバランスのよいプログラミングが可能になる。プログラムのテストは完成した上位モジュールから行うのが常で、これをトップダウンテストという。
(類)段階的詳細化

トップメニュー[top menu]

アプリケーション、WWWページなどの最上位のメニュー。

飛越し[jump/GOTO]

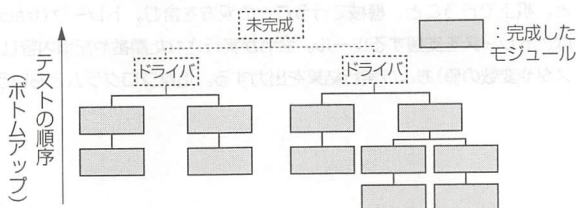
プログラムのある部分から他の部分へ制御を移すこと。飛越しを指示する命令を飛越し命令、分岐命令という。プログラム言語により指定法は異なるが、代表的な飛び越しには、次のような例がある。

(1)**無条件飛越し**：制御をプログラムの別の場所へ条件コードや状態語とは無関係に移すこと。無条件分岐命令ともいう。goto文などによる。単に飛越しといえば無条件飛越しをさす。(参)基本制御構造

(2)**脱出**：ループ内の処理を終え、制御をひとつ外へ移すこと。break文、continue文などによる。(関)繰返し

(3)**復帰**：return文などによる。関数やサブルーチンから呼出し元への戻り。(関)戻り番地、戻り値

ドライバ[driver]

ボトムアップテストで下位レベルのモジュールまたはモジュールの集まりをテストするために、その上位レベルの未完成モジュールの代替として用いるテスト用のプログラム。テストする下位のモジュールを呼び出して、テストデータを渡す役割をになう。
(参)ボトムアップテスト、(対)スタブ

トラック[track]

記憶媒体がひとつの読取り部または書込み部(ヘッド)を通過する際に、そのひとつの読取り部、書込み部に対応する媒体上の記録帯。磁気ディスクの場合はアクセスアームを移動することなく、情報を読み書きできる部分で、同心円状になる。光ディスクでは渦巻き状になる。複数の磁気ヘッドを用いる磁気テープでは平行な9〜18本のトラックがある。(関)セクタ

トラック索引[track index]　(関)索引順次編成

トランザクション[transaction]

データファイルの内容に、増減、変更を行うすべての取引で発生するデータ。マスタファイルを更新するためのデータ、またはその仕事。オンラインシステムにおけるトランザクションは、通常いくつかの入力メッセージをもち、その応答としての出力メッセージを生み出す。

トランザクションファイル[transaction file]

業務の遂行にともなって新規に発生した一時的なデータを含むファイル。適用業務に応じて、しかるべきマスタファイル(基本ファイル)と突き合わして処理する。マスタファイルに対して入力ファイルとなる。このファイルもまた内容を更新しながら継続して用いる。一時ファイル、変動ファイル、発生ファイルともいう。

トランスレータ[translator]

①翻訳プログラムの総称。(同)翻訳プログラム

②Cでは、コンパイラ以外の翻訳を行うプログラム。特にプリプロセッサを指すことが多い。トランスレータの処理は対象がテキストで、処理系が定義した文字表記やエスケープシーケンスを文字に変換したり、コメントの削除、プリプロセッサトークンの抽出と字句トークンへの変換、ヘッダのインクルード、マクロの展開等の処理を行う。この結果が翻訳単位となり、コンパイラに渡される。(同)プリプロセッサ

トレース[trace]

開発中のプログラムをデバッグする目的で、コンピュータが実行した命令と、その実行によってレジスタの値や主記憶の内容などが変化していく過程を1ステップ実行するごとに追跡し、制御構造や分岐を検査すること。机上で行うこと、機械で行うことの双方を含む。トレーサ(tracer)は、トレースを支援するツール。命令が実行された順番や記憶内容(レジスタや変数の値)および実行結果を出力する。追跡プログラムともいう。

内部スキーマ[internal schema] （参）スキーマ

内部設計[detail design]

外部設計（概要設計）をもとに行う、システムの詳細な設計段階。ファイルの仕様、プログラムの構造、機能、入出力帳票や処理手順の具体的で詳細な設計。例えば、入力する原始伝票の設計やプログラム設計書の作成などがある。設計情報（information for system design）は、プログラムの内部設計の段階で管理する情報の集合。多くの技術者が設計作業にかかわるため、また検討の結果、変更されることも多いので、常に最新の仕様が一元的に管理され、技術者の端末から登録、修正、照会できることが望ましい。おもな内容は、（1）データ定義情報、（2）データ項目、（3）プログラム情報、（4）ファイルレイアウト情報などである。

内部ソート[internal sort]

ソート処理の過程で、入力ファイルから主記憶装置の容量が許す範囲のデータを読み込んで主記憶装置内部で並べかえを行う作業。ソートの手法にはさまざまなアルゴリズムが考案されている。

（例）バブルソート、クイックソート、（対）外部ソート

内部表現[internal representation of data]

データやプログラムの主記憶上での表現方法。内部形式ともいう。内部表現は、実行環境によって異なる。

内部ラベル[internal label] （関）ラベル

内部割込み[internal interruption] （関）割込み

流れ図[flowchart] （同）フローチャート

ナノ[nano]

10^{-9}（10億分の 1）を表す補助単位。ナノ秒はコンピュータ内部の 1 命令実行時間を表すときに用いる単位。

なりすまし[pretense]

公開かぎ暗号化方式で、正規の利用者に気づかれずに公開かぎをすりかえること。これによって、本来の秘密かぎをもった利用者は暗号を解読できず、すりかえた人間の秘密かぎで解読ができるようになる。公開かぎをもった人が気づかないうちに、別の人間が本来の秘密かぎをもった人になりすましてデータの盗聴を行い、結果的に犯罪すらできることなる。なりすましの対策として、公開かぎがすりかえられたものでないことを証明する仕組みが必要となる。すなわち、第三者の認証機関を設けて公開かぎ証明書（発行局の電子署名付き）を発行してもらう。公開かぎが登録されたものであるか認証機関（証明機関）に確認することによって、なりすましを防

ぐことができる。(参)PEM、認証、証明機関

ナル[null]

空文字。媒体の空きや時間の空きを埋めるための制御文字。挿入しても削除しても文字列の意味は変わらない。装置制御や情報形式にも影響を与えない。ナルストリング(ヌルストリング)は何も含まない長さゼロの文字列である。Cでは全ビット0で表現される制御文字のひとつ。文字列の終わりに付加される。

ニ

二項演算子[binary arithmetic operators]

算術演算子のひとつ。データ項目や定数あるいは括弧でくくった算術式の間に書き、それらの項目について四則演算などを行う。これには、+(加算)、−(減算)、*(乗算)、/(除算)、**(べき乗)がある。

(関)算術演算子

2進数[binary numbers]

2進法で用いる数字、または2進法を用いて表した数値。

2進法[binary notation]

2を基数とする数の記数法。数字として 0、1の二つを使う。コンピュータ内部で数値を取り扱うときの表し方の代表的な形式。回路などで直接的に電圧の高低を示す場合には、0、1に代わりL(low)、H(high)などの記号を用いることもある。2進数の最小表現単位がビットである。

2層スキーマ[two schema structure]

CODASYL仕様のネットワーク型データベースでは、スキーマを2層でとらえ、論理ファイルの定義をサブスキーマ、物理ファイルの定義を記憶スキーマという。スキーマがデータベースの論理構造を全体として把握したもので、管理者の見方に近いのに対し、サブスキーマはスキーマの一部であって、適用業務プログラムにデータベースの構造を見せるプログラマの見方に近い。この役割をさらに分化させたのが3層スキーマである。

2の補数[two's complement]

n桁の2進数の「2の補数」とは、2nからその数を引いた数をいう。「1の補数」に1を加えた数と同じ値となる。

2バイト文字コード[double byte character code : DBCC]

2バイトで構成する文字。日本ではJIS情報交換用漢字符号が代表的。このほか、中国、日本、韓国などの漢字文化圏で用いるCJK統合漢字コード、EUC(拡張UNIXコード)、ISOが規定するUCS-2が2バイトコードで

ある。(参)JIS情報交換用漢字符号、UNICODE

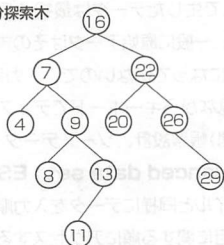
2分木[binary tree]

木構造のひとつ。根(ルート)と呼ばれる根幹となる端点から二つ以下の枝がでるもの。各端点は節と呼ばれ、枝が0の端点は葉と呼ばれる。(参)木構造

2分探索[binary search/dichotomizing search]

2分木を利用した探索法。配列(リスト)中のレコードがキー項目の大小順(昇順または降順)に並んでいるときに効率のよい探索法のひとつ。2等分探索法、2分割法ともいう。原理的には、レコードの並びのほぼ中央に位置するデータとソートキーの条件を比較して探索範囲を2分し、レコードの存在するはずのない一方を探索範囲から除外する。これを繰り返し少ない操作で効率よくデータを探す手法である。データの分布に偏りがあり、分布が既知の場合は、効率のよい別の分割法を用いることもある。AVL探索法やフィボナッチ探索法などがある。(関)探索

2分探索木[binary search tree]

2分木の頂点にレコードを入れ、右側の枝の先の頂点(節)のすべては上位の頂点のデータより小さいものがあり、逆に左側には大きいデータがあるようにした木構造のこと。2分探索と同様の原理で、目的のデータを探すことができる。

日本語フロントエンドプロセッサ[front end processor for kana to kanji transformation:FEP]

パソコンなどのシステムに日本語を効率的に入力するためのソフトウェア。日本語ワープロソフトのローマ字かな変換、かな漢字変換機能だけを独立させたもの。単にフロントエンドプロセッサ(FEP)と略す場合もある。メモリに常駐させて使用し他のアプリケーションソフトに組み込む。

入出力設計[input-output design]

外部設計の際に、コンピュータシステムにデータを入力・出力する方法を設計すること。開発部門は、データの用途、目的を十分に考慮し、業務

部門の協力を得て設計する。入出力の媒体・タイミング・分量・入手法・配布法などを検討する。特に入出力のヒューマンインタフェースには注意を払い、入力帳票(伝票)と画面レイアウトの設計は見やすさ、操作のしやすさを考慮して、データ項目を配置し画面定義する。また出力帳票レイアウトの設計では、タイトル・項目の編集、改ページ、集計単位などの処理条件等をよく考えて定義する。(関)帳票設計、画面設計、報告書設計

入出力装置[input-output unit]

コンピュータシステムを構成する入力装置と出力装置の両方を合わせて入出力装置という。システムとユーザの接点に当たる機器。パソコン等の普及によりコンピュータの操作者が専門技術者にとどまらなくなって、ヒューマンインタフェースの重要性が増し、現在でも改良、改善を進めている。(例)入力装置、出力装置、(参)ヒューマンインタフェース

ニューメリックチェック[numeric check]

入力データのチェックの一手法。数値項目と定義された項目が数字であるかどうかを調べる方法。数字検査。通常、数字項目は計算処理の対象となり、誤って英字やカナ文字が入るとエラーの原因となる可能性があるため、この検査を行う。

入力原票[source document]

コンピュータ処理の対象とする原始データ(source data)を記載した帳票。例えば、取引が発生することで生じたデータは最初は伝票に記録する。この伝票が原始データである。一般に原始データはそのままの形式ではコンピュータ処理に適した形式になっていないので、入力原票(原始帳票=原票)とディスプレイ画面を見ながらキーボードでディスクなどの入力媒体に変換することが多い。(関)帳票設計、ソースデータ

入力順データセット[entry sequenced data set：ESDS]

仮想記憶において、順編成ファイルと同様にデータを入力順に記憶する編成をもつデータセット。キー項目に関する順にアクセスする場合は、入力データをあらかじめキー順に整列しておく必要がある。ESDSに対するアクセスは、基本的には順次アクセスであり、先頭レコードからの正順、もしくは最終レコードからの逆順のアクセスとなる。ただし、各レコードの相対バイトアドレスを利用者が認識していれば、直接アクセスすることも可能である。

ESDSに対する処理を行う場合、更新に関しては更新するレコードが元のレコードと同じレコード長である場合は可能である。レコードの追加は、最終レコードの後ろに追加する場合に限り可能である。削除は、物理的な削除ではなく削除マークを付ける形では可能である。その他の場合は、新しくファイルを作り直す必要がある。

入力設計[input design]

データ入力時の作業に関する仕様を決定すること。入力するデータの項目や量を考えて原始データの帳票、入力装置の画面やキーボードの使い方をくふうし、入力者の使いやすさ、効率性を考えて設計する。入力エラーがあったときの指摘法や再入力のタイミングなども考慮する。

入力装置[input unit/input device]

コンピュータシステムを構成する装置で、システムにソースデータ、ソースプログラム、指示(コマンド)などのデータをシステム外部から与えるために用いる装置の総称。データをコンピュータ処理に適した形式の電気信号に変換し、記憶装置に送る。例えば、キーボード、補助記憶装置などがある。

ニル[nil/NIL]

ポインタの一種で、どこも指定しないポインタ。(参)ポインタ

認証[authentication]

データまたは文書が一定の行為、正当な手続きおよび方法、正当な人でなされたことをオンラインデータ上で証明するセキュリティ技術。

書類上での認証はサインや印鑑で行うが、オンラインデータはサインや印鑑をそのまま入力することはできない。また、データそのものの改ざんや不正使用も故意か不注意か別にして、特権さえあれば本人でなくても可能である。そのため、オンラインデータの認証を行うソフトウェアがある。認証ソフトウェアは、オンラインデータの認証に公開かぎ暗号化技術を応用し、ユーザがそのデータや文書にアクセスできる本人かどうかを証明する。(参)証明機関、S/MIME、PEM、なりすまし

ネ

ネスト[nest]　(同)入れ子

ネットワーク[network]

広義には、情報の連絡網・通信網で人材やグループを含む。狭義には通信ネットワーク、コンピュータネットワークをさす。「網」と訳す場合も多い。物理的、論理的に機器やデータを網状に連結した構造、システムをさす。

①情報通信の分野では、端末を結ぶ機器(交換機など)と伝送路で構成するデータ伝送用の通信媒体の集合。この場合、特に通信ネットワークという。(同)通信ネットワーク

②情報処理の分野では、通信ネットワークを利用したオンラインシステム、分散処理システム。この場合、特にコンピュータネットワーク、計算機ネットワークという。

③要素の関係を表すデータ構造の用語では、あるデータが複数の上位データ、複数の下位データに関係づけられる構造をさす。

(同)ネットワーク型データベース

ネットワーク型データベース[network structure database]

構造型データベースうち、レコード間の関係が、どのレコードどうしも関連づけることができるネットワーク型(上位と下位の関係がm：n)のデータモデルに基づいたデータベース。CODASYLのDBTGが提案したものが代表的で、CODASYL型データベースともいう。

ネットワーク管理[network management]

ノ

ネットワークの信頼性を高め、効率的に使用するために行う管理。管理対象は、通信回線や交換機のみならず、ネットワークに接続され情報処理を行うコンピュータや端末にまで及ぶ。管理項目には次のものがある。

(1)ネットワークを構成しているあらゆる構成要素の管理(構成管理)。

(2)障害の早期発見と告知、原因の切り分けと修復などの実施(障害管理)。

(3)システムの性能データを収集、評価(性能管理)。

(4)資源の不正使用の有無を確認、アクセス管理などの実施(機密管理)。

(5)利用者への影響を極力抑えながらの構成変更作業(構成変更管理)。

(6)利用者の利用量に応じた料金の請求(課金管理)。

(7)利用者の操作ミスを少なくして不要な負荷を抑制(操作管理)。

ネットワークの規模と複雑性が増大するにしたがって、ネットワーク管理の重要性も高まる。また、人間の管理能力を超える管理レベルを要求するほどシステムが巨大化した場合は、管理の自動化を含めた管理システムの構築が必要となる。

ノ

ノード[node]

広義には、ネットワークを構成する要素を論理的なモデルとしてとらえたときの呼び方。(参)節、木構造

ノンストップコンピュータ[non-stop computer]

(同)無停止コンピュータ

バージョン[version]

プログラムやマニュアルなどの版番のこと。Version3.0などとバージョンナンバーを付けて、ソフトを更新するごとに版番を上げ、改良と向上の度合を示す。通常、最初の公開ソフトをVersion1.0、それ以後ソフトを更新するごとにVersion2.0、Ver2.1などと表示する。版番を示す数字のつけ方には特に規則はないが、部分的な小改訂では0.1を加え、機能を刷新した場合には1.0を加えることが多い。

バージョン管理[version management]

プログラムやファイル、マニュアルなどのバージョンを管理し、システム運用中の混乱を招かないようにすること。更新にともなう情報やファイルの管理。運用中のシステムでは、プログラムのバージョンアップやファイルのアップデートを繰り返しながら長期間にわたり利用することが多い。混同は事故や障害を招く場合があり、常に最新のバージョンを利用するべきである。いっぽう万一の事故に備え、更新前のプログラムやファイルを保存しておく必要もあり、多数のファイル情報を正しく管理、利用しなければならない。

ハードコピー[hard copy]

コンピュータの出力情報のうち、人間が直接読むことができ、持ち運びできる情報。通常、用紙に印刷した情報をさす。(対)ソフトコピー

ハードディスク装置[hard disk unit：HD]

ハードディスクを記憶媒体とする磁気ディスク記憶装置。通常は固定式で、フロッピーディスクのように媒体を交換することはできない。フロッピーディスク装置に比べアクセス時間が短く、記憶容量も大きい。汎用コンピュータ用大型装置からパソコン用の小型装置まで多くのクラスがある。全体的に大容量化、小型化、高速化が進んでおり、パソコンでも数ギガバイト級のハードディスク装置を内蔵するようになった。インタフェース規格としては、SCSI、EIDE、ATA-33／66などがある。

倍精度[double precision number]

実数を表現するデータ型で、標準の浮動小数点数以上の精度で、多倍精度以下の精度で数値を表現する。二倍精度、長精度ともいう。1ビットの符号ビットと、指数部、仮数部から構成されており、その桁数は標準精度の2倍を使用している。(関)単精度

排他制御[exclusive control]

同時に共有することが不可能な資源をいくつかのプロセスが使用したいとき、資源をめぐる競合、衝突が生じないように、ある時間帯にはひとつ

のプログラムだけに使用を許可すること。共有不可能資源には、CPU時間、マスタファイル、共用メモリ、通信回線などがある。複数の共有不可能な資源の排他制御では、各プログラムの資源排他制御順序によってはデッドロックになる危険性もある。このため、あるプログラムでデータを処理する前に、そのデータを他から参照しないようにロックし(lock＝かぎをかけ)、他のプログラムによる更新を禁止する。処理終了後、アンロック(unlock＝かぎを解除)し、待たせていた他のプログラムでの使用を許可する。(関)デッドロック

排他的論理和[exclusive disjunction／exclusive OR：EOR／XOR／EXOR]　(同)EOR

バイト[byte：B]

①一般には、8ビットの列で表したデータの単位。8桁の2進数に相当する。これにより256個の組合せが可能となり、数字、アルファベット、記号など基本的な文字を表現できる。特殊な場合、7ビットを1バイトとし、7ビットバイトまたはセプテットという。

②データの大きさ、主記憶や補助記憶の記憶容量の単位。文字数に置き換えれば、ANK文字なら1バイトが1文字に、漢字なら2バイトが1文字に相当する。

バイナリ[binary]

2進、2分、2値などを意味する語。

バイナリサーチ[binary search]　(同)2分探索

バイナリセーブ[binary code save]　(関)セーブ

バイナリファイル[binary file]

①広義には、標準の情報交換用符号によらないファイル。この意味では内容はテキストであっても、暗号ファイル、画像や文書の圧縮ファイル、アプリケーション独自のデータファイルもバイナリファイルである。

②狭義には、システムの内部表現の形式(機械語や2進コード)のまま記録したプログラムやデータのファイル。入出力時の変換を必要としないため、アクセス時間が最小限ですみ、一般的には記録に要する領域も少ないが、その内容は異機種間での利用は難しく、データであってもコード変換等の作業を必要とする。また、情報交換用符号などのような標準的なコード化をしていないため、内容をそのまま表示しても意味を読み取ることができない。

ハイパーテキスト[hyper text]

情報を、項目ごとに整理し論理的なリンク(関連付け)構造を与えた文書データ。最初に表示した画面から別の画面を次々に表示させるなど、今までのテキストファイルとは異なる利用分野を形成した。先頭から順に読む

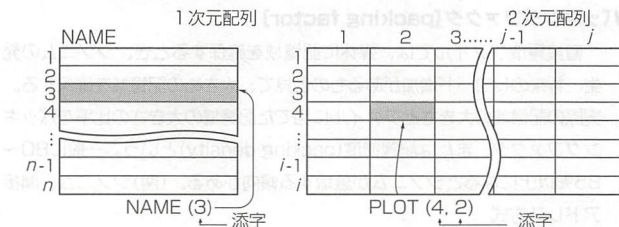
のではなく、百科事典や六法全書のように目的の記述個所を探して読むような文書をシステム上で実現したものである。電子ブックがその例で、文書データに、図形、画像、音声などの情報を含ませることもできる。マルチメディア対応のドキュメントシステムである。

制作用のソフトとしてMacintoshのHyperCard、ページ記述言語のHTMLなどがある。多くのWWWサーバのコンテンツはハイパーテキストの形で提供されている。なお、ハイパーテキストを読むソフトをナビゲーションソフトまたはブラウザという。

ハイパーリンク[hyper link]

WWWページ上の文字をクリックすると、あらかじめ指定した別のページやアンカー(一種のラベル)の位置などにジャンプする機能。

配列[array]

有限個の同じ型の変数の要素を表の形に並べたもの。各要素は、配列名とその順序に付けた添字により参照が可能である。主記憶装置内では、連続した記憶領域内に番地順に記憶される。要素を指定する添字の個数により、1次元配列、2次元配列、3次元配列、……という。1次元の配列は論理的構造と主記憶装置内での物理的並びが一致しており、順次処理で用いることが多い。多次元の配列でも処理系によって、主記憶装置内の1次元の並びに置き換えられる。(関)テーブル

バグ[bug]

意図とは異なる処理、誤動作、暴走などの原因となるプログラム中の誤り。バグには形式的な誤りと論理的な誤りがある。虫ともいう。
(参)デバッグ、デバッグツール

パス[path]

ファイルをディレクトリという単位で管理する階層型ファイルシステムでは、ファイルはディレクトリの階層の下に存在する。したがって、特定のファイルを見つけるためには、いくつかのディレクトリをたどる必要がある。この検索ルートをパス(経路)という。ルートディレクトリ(最上位)からのパスを絶対パスといい、カレントディレクトリからのパスを相対パスと呼ぶ。(参)ディレクトリ、ファイル管理システム

パスワード[password]

コンピュータ、端末、回線、ファイル等の正当なアクセス権を有する利用者本人かどうかを識別する文字列。セキュリティ対策の最も一般的な方式で、システム使用者の認定を行うために、本人しか知らない記号列をあらかじめシステムに設定しておく。システムの使用者は、ログオン時にまず利用者名(ユーザID)を入れ、それからパスワードを入力する。

(1)端末やパソコンがLANに接続している場合には、ローカルマシンのパスワードが必要である。

(2)ユーザがネットワーク(ドメイン)へログオンする際に使用するパスワードはネットワークパスワードという。これは、サーバ側で一元管理しており、パスワードを変更すると、サーバ側の情報を書き換える。

ハッカー[hacker]

ハ

原義(手斧一丁でログハウスを作る人)から転じて、コンピュータ上で人が驚くような大きな仕事ができるプログラマの尊称。さらに原義から遠ざかり、コンピュータにかじりついて離れないマニア的技術者をさす語になった。非合法にコンピュータシステムに侵入し、プログラムやデータの盗用、改ざん、破壊などの悪質な罪を犯す者をさすことがある。しかし、システムに対する犯罪者はクラッカーまたはブレイカーと呼んで区別するべきだという主張もある。

パッキングファクタ[packing factor]

直接編成ファイルでは、媒体に記憶域を確保するとき、シノニムの発生、将来のレコード増加があるものと見て、大きめの記憶域を確保する。実際の記憶域の大きさとファイルに当てた記憶域の大きさの比率をパッキングファクタ、または記録密度(packing density)という。一般に80~85%以上になるとシノニムが急増する傾向がある。(関)シノニム、間接アドレス方式

パック[pack]

ゾーン形式の数値を演算の際にパック形式に変換すること。逆にパック形式のデータをゾーン形式に変換することをアンパックという。

(対)アンパック

バックアップ[back up]

システムの障害に備えて、代行できるシステムを用意すること、また実際に代行すること。主としてハードウェア面では待機システムをソフトウェア面ではその複製をさす。

①記憶媒体で供給されたパッケージソフトについてのバックアップは、次のようなことがらをさし、単に複製することではない。

(1)コンピュータシステムに障害が発生し、再度インストールしなけれ

ばならないときに備えて、原典または複製を保管すること。

(2) 常時使用するシステムディスクを誤って破壊した場合に備えて、原典を使用せず、複製を使用させるための複製。

(3) ファイルやプログラムの更新に先だって、更新したシステムに完全に移行するまで、現在使用中の資源や環境を保存すること。

②高い信頼性を要求するシステムでは、可能な限りシステムを多重化すること。障害が発生した場合、短時間に切り換えて代行システムを使用する。

(1)バックアップファイル：ファイル障害が発生した場合は、ハードウェア上の障害ならディスクの交換などを行ったあと、最新のバックアップファイルの内容をコピーする。そのあと、バックアップファイルを作成した日時以降、障害が発生するまでのジャーナルログを使用して復旧する。トータルセーブファイルともいう。

(2)バックアップコンピュータ：長時間のシステムダウンが許されないような高信頼性システムで使用する。デュプレックス構成システムでは、待機系コンピュータともいう。EUCでは、ハードウェアの故障中、修理中に使用できる代替のコンピュータをさすこともある。

(3)バックアップ回線：ネットワークの迂回ルート。例えば、専用回線に障害が発生して使用不能になったとき、ISDNをバックアップ回線として切り替えるというような方法をとる。障害回復までは、スピードが多少遅くなったり、料金を別にとられることがあるが、まったく使用できなくなるという最悪の事態は避けられる。

バックエンドプロセッサ[back end processor：BEP]

①複数のプロセッサからなる機能分散型のシステム(タンデムシステムなど)で、通信制御などの前処理を分担するフロントエンドプロセッサ(FEP)に対し、主処理後のファイル処理などを行うプロセッサをバックエンドプロセッサ(BEP)という。

②メイン処理とは別にデータベース処理専用のプロセッサなどを接続している場合は、それをバックエンドプロセッサという。FEPやBEPは相互に独立して処理を行い、磁気ディスクなどを介して情報交換をする。

バックグラウンド処理[background processing]

タイムシェアリングシステムや多重処理システムの環境のもとで動くコンピュータは、平常時には余剰能力をもつ。この空き時間を利用して、低優先順位でコンピュータを活用する処理。例えば、言語処理、バッチ処理、プログラム開発などに使う。パソコンなどでは、ワープロソフトで文書編集を行うバックグラウンドで印刷処理を実行する。(対)フォアグラウンド処理

パック10進数[packed decimal number]

パック形式(pack format)で表現した10進数。10進数の各桁を4ビッ

トのBCDコードで表し、最下位4ビットで符号(通常プラスは1100、マイナスは1101である)を示す形式。内部10進、パック10進ともいう。(対)ゾーン形式、アンパック10進数

パッケージソフト[package software]

個々のユーザの異なる用途のために個別に受注し開発したソフトではなく、ある機種に合わせて多くの人が同じ目的で使えるように考慮して商品化したソフト。(類)ソフトウェアパッケージ

ハッシュ関数[hash function]

ハッシュ法に用いる関数。ハッシュ関数は、キーの分布の偏りを考慮したうえで、ハッシュ表に無駄を作らず、しかもできるだけ均等に配分する関数が望ましい。例えば、キーが整数で均等に分布していれば、mで割った余りは0からm−1までの整数になり、m個のグループに均等に配分できることになる。(関)ハッシング

キー
113 → 演算 → 格納領域 = 結果3
ハッシュ関数
01 12のデータ
23のデータ
02 24のデータ
46のデータ
あふれ域
03 → 69のデータ
86のデータ
113のデータ
04 60のデータ
⋮ ⋮

ハッシュ合計[hash total]

データチェックの一手法。レコードまたはファイル中の特定の複数の項目の和で、その合計自体は特に意味をもたない。入力前に人手で算出し、データとともに入力し、システム側の算出値と比較することで入力データの誤りの有無を検査する。例えば、同一レコード内の金額とコード番号の和、レコードごとの個人番号の和など。

ハッシュ法[hashing/scatter storage method]

ハッシュ表とハッシュ関数を用いたレコード探索法のひとつ。データを記憶する段階で、キー値から演算により格納場所を算出し、その場所に記憶させる。データを探索する際には、キー値に同じ演算を施せば格納場所が得られるという高速の探索法である。

問題点は、演算のルールしだいでデータ量より多くの格納場所をあらかじめ確保しなければならないこと(記憶装置の資源の無駄がある)、異なるキーの演算結果が一致して(衝突またはシノニムの発生という)、記憶場所の競合が起こり得ることである。競合する場合は、同じ場所にあてがわれたレコードをポインタでつないでおく。これをチェーンという。

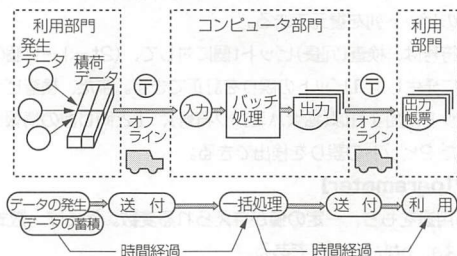
ハッシング[hashing/hash addressing]

直接編成ファイルで、レコードのキーからレコードアドレスを定める方

法にハッシュ法を用いること。レコードのキーとレコードアドレスは1対1で対応すれば理想であるが、一般的には、重複(シノニム)ができたり、空き領域ができたりして記憶装置の使用効率が低下する。そこで多対1の対応でアドレスを求め、同じアドレスに当たったレコード群をバケット(bucket)に収納する方法をとる。このバケットに割り当てられるレコード数をできるだけ一様にするため、ハッシュ関数を用いる。

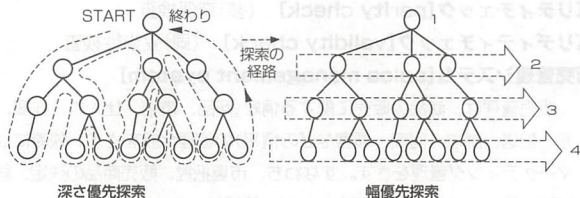
バッチ処理[batch processing]

データ処理の一形態で、データをある一定の分量、または一定の時点になるまでためておき、一度にまとめて順に処理する方式。一括処理方式ともいう。バッチ処理は処理手順が確定しており、処理の対象となるデータもファイルの形式ですべてそろっている場合に適用される。例えば、毎月1回の給与計算、各種料金の支払いなどの月次処理、企業会計などの年次処理、また、全ソースデータが揃った時点とか、在庫数が定めた発注ロットに達した時点の処理はバッチ処理方式で処理する。

(関)リモートバッチシステム

バッファ記憶装置[buffer storage]

データ処理システムの構成要素間の処理速度の違い、情報要素間の事象の発生時間を補正(緩衝)するために用いる記憶装置。例えば、中央処理装置と主記憶装置の間におくキャッシュメモリ、入出力装置と主記憶装置の間にあって、両者を独立に動作させるためのディスクキャッシュなど。

幅優先探索[breadth first search]

木構造のグラフのデータ探索法のひとつで、ルートからスタートし、同

一の深さにあるすべての頂点(節)を探索し、なければ次第に深いレベルの頂点を調べる探索法。探索に成功したときは、最短経路が記憶されている。最も深い葉(木の高さのレベル)に至ってもデータがなければ失敗である。
(対)深さ優先探索

バブルソート[bubble sort]

最も簡単な内部ソートのアルゴリズムのひとつ。配列中のデータの隣り合った二つを比較し、逆順であれば入れ換える。これを何回も繰返し、ソートを完了する。データ量が多くなるとその2乗に比例する計算量が必要で、効率のよいアルゴリズムとはいえない。隣接交換法ともいう。

ハミングコード[Hamming code]

誤り訂正符号の一種。この符号方式を考案したR.W.Hammingの名前をとって名づけた。これにより誤りの検出と訂正ができる。基本原理は、送信側が検査ビット(パリティビット)を元のビット列のある部分と組み合わせると一定の規則になるような位置に埋め込む。受信側がその規則と整合性をチェックすると、誤りのあったビットの位置がわかり、そこを反転すれば元のビット列を復元できる。

この符号は、検査(冗長)ビットt個に対して、(2t−1−t)個の情報ビットの中に発生した1ビットの誤りを訂正できる。また、検査ビットにさらに1ビット付加すれば(拡大ハミング符号)、上と同じ数の情報ビット中に発生した2ビットの誤りを検出できる。

パラメタ[parameter]

①ある用途をもち、一定の値が与えられる変数。例えば、数式 y=ax+bではa、bがパラメタである。

②プログラム内で、手続きや関数の内容を定義する変数で、プログラム間の連絡をとるために、別のプログラムに引き渡す値。ALGOL、BASIC、Cの用語で引数と同義。(参)引数

バランスチェック[balance check]

もとは会計処理の貸方合計と借方合計の一致などをチェックする方法。転じて二つのデータが均衡しているか調べて、誤りの有無をチェックすること。平衡検査ともいう。

パリティチェック[parity check] (参)奇偶検査

バリディティチェック[validity check] (同)妥当性検査

販売管理システム[sales management system]

販売管理は、商品の販売に関する情報(商品、市場、仕入れ、在庫、販売、配送、請求、回収、顧客などの情報)の管理全般をさす。狭義には、マーケティング管理をさす。すなわち、市場把握、販売商品の決定、販売組織の確立など販売力の強化をめざす管理である。

(1) POS(point of sales)：電子式金銭レジスタ、バーコード読取り装置、クレジットカードの自動判別装置などの機器により収集した商品の販売情報などをコンピュータに送り、情報の有効利用を図るためのシステム。

(2) EOS(electronic ordering system)：受発注業務を自動化したシステム。POSシステムと並び、商品の迅速な仕入れ、販売活動を支援する。顧客のニーズの多様化により、商品の多品種少量生産が進み、受発注業務が煩雑化する傾向にある。これをコンピュータネットワークを利用することで簡便化、高速化を図るねらいがある。

番兵[sentinel]

線形探索で探索の終了時を検出するための手法のひとつ。配列(リスト)の末尾にあらかじめ、探索するキーと同じ値(番兵)を入れておき、探索中、キーが一致すれば探索を終了する。探索に失敗した場合でも、最後には探索を終了する。番兵を設けると、データ件数を数えて終了時を判断するルーチンが不要になる。

汎用プログラム言語[versatile programming language]

プログラム言語のうち、比較的広い分野で多目的に使用できる仕様のプログラム言語。アセンブラやBASIC、Cなどの高水準言語をはじめ、多種多様なものがある。特殊問題向き言語に対比する語。

汎用レジスタ[general register／general purpose register：GR／GPR]

中央処理装置内の番号で指定が可能なレジスタ群。通常、累算器や指標レジスタ、データの特殊操作用など、多目的に使う。数個～数十個程度のレジスタで構成し、プログラム内でさまざまな種類のレジスタとして指定できる。

ヒ

比較演算子[relational operator]

プログラムやマクロ命令中で、数字や文字の大小比較をする条件などを指定するとき、二つのオペランドの比較を関係式に書くための演算子。if

比較関係	数学記号	演算子の例		
小さい	＜	.LT.	LESS THAN	<
小さいか等しい	≦	.LE.	NOT GREATER THAN	<=
等しい	＝	.EQ.	EQUAL TO	=, ==
等しくない	≠	.NE.	NOT EQUAL TO	!=
大きい	＞	.GT.	GREATER THAN	>
大きいか等しい	≧	.GE.	NOT LESS THAN	>=

文の条件判断に使用する。比較できるのは同じ型のデータどうしである。プログラム言語等によって異なった記号を用いる。関係演算子、比較作用素ともいう。ほとんどのプログラム言語、表計算ソフトでは以下のような演算子で比較ができる。比較演算の結果が真の場合、偽の場合にとる値は処理系によって異なるが、多くは非0、または0の値をとる。

比較命令[compare instruction]

コンピュータ内部で二つのデータ項目の値の大小(数値の大きさだけでなく、文字列の前後関係などを含む)を比較する命令。また、比較の結果により、コンディションコードを設定する。計算機プログラムの実行中に、複数の命令からひとつを選ぶように指示する分岐命令(branch instruction)の実行に先だって実行される。演算の結果により条件コードの値をセットし、その値により飛越しを発生させる。判断命令、条件つき飛越し命令と同義。条件コードによらない分岐命令は無条件分岐命令(飛越し命令)という。

ヒ

引数[argument]

サブルーチンや関数副プログラムを呼び出すときにプログラム内で引き渡すデータやパラメタ。例えばFortranで、xを与えて平方根を求めるのに、プログラム側では組込み関数SQRTを用いてSQRT(x)と表現する。このxを引数という。関数側ではxを受取って処理を行い、の値を戻り値としてプログラムに返す。

(1)実引数(actual argument)：関数や副プログラムを呼ぶプログラム側から関数側に渡す引数。実引数は関数の中で仮引数に代入される。

(2)仮引数(dummy argument)：手続きや関数の呼出し元のプログラムから実引数を受け取る引数。仮パラメタとも呼ぶ。

引渡しテスト[delivery test]　(類)受入れテスト

ピコ[pico：p]

10^{-12}(1兆分の1)を表す補助単位。

非実行文[non-executable statement]

データの型や値を宣言したり、関数を定義したり、記憶領域を確保したりするなど、処理を行わせる方法を定義する文。例えば、COBOL言語では、見出し部、環境部およびデータ部の記述が、Fortranでは宣言文、DATA文などが非実行文である。(対)実行文

非常駐プログラム[non-resident program]

コンピュータシステムで、システムが必要とした場合に、補助記憶装置から呼び出して使用するプログラム。(対)常駐プログラム

非数値データ[non-numerical data]

コンピュータシステムがコードで扱うデータのうち、数値データでない

データ。文字データと論理データがある。なお、画像データ、音声データなどはコードで扱うデータでないため、非数値データとは言わない。

ビッグバンテスト[big bang test]

すべてのモジュールが完成し、モジュールテスト(単体テスト)が終了した後、全モジュールを結合したうえで行うテスト。非増加テストともいう。未完成モジュールがあればドライバやスタブを用意する。単体テストは終了しており、テストの対象はモジュール間インタフェースに限定できるが、モジュールの数が多いと原因追及の負担が大きい。大規模システムでは、この前段階で結合テストを行う。ビッグバンテストは比較的小規模なシステムのテストに適している。

ビット[bit:b]

2進法の数字1桁(binary digit)の略。情報量の最小単位。1個の2進数字が保有する情報量を表す。2進数の1桁に相当し、2値のいずれかをとることで他と区別する最小の情報量である。 多くの情報を扱うために何ビットかを組み合わせたビット列を1語とする。一般にnビットを組み合わせた語(n桁の2進数相当)では、2n個の情報を区別、表現できる。8ビットを単位として1バイトという。

(参)バイト

否定[negation/NOT]　(同)NOT

非手続き型言語[non-procedural language]

プログラム記述の命令の順序とハードウェアのプログラム実行順序が強い関係をもたないプログラム言語。COBOLやFortranなど初期の高水準言語は、ハードウェアの実行順序にそって処理の流れ(手続き)を記述する形式に偏っている。1980年代から新しいプログラミング技法を取り入れた非手続き型言語が登場した。

(例)Lisp、Prolog、Smalltalk、SQL、表計算ソフト

非ブロック化[deblocking]　(同)デブロッキング

ビュー[view]

広義にはデータベース利用者のデータに対する見かた(データの論理的な構造)。データベースシステムの記憶装置内のデータの扱い(データの物理的な構造)そのものは、すべての人に開示できない、人間にわかりにくいなどの理由で、利用者に見せるための機能や操作法が定めてあり、これをビューということもある。システム固有の定義によって異なり、ネットワーク型データベースでは外部スキーマの一部であり、関係データベースでは、関係演算の結果としての関係がビューである。

ヒューマンインタフェース[human interface:HI]

人とコンピュータシステムとの間を結ぶ入出力機器や入出力方式。

ヒ

(1)人間とシステムの間の対話を行えるような装置で、各種OA機器、VDT機器などがこれに相当する。従来マンマシンインタフェースと呼ばれていた。
(2)人がそれを操作することでシステムに意思を伝え、またシステムは表示装置などに応答・結果等を表示すること。特にエンドユーザに使いやすいコンピュータシステムをめざすうえで、ハードウェア面、ソフトウェア面を含め表示方式や入力方式をくふうすることは、ユーザインタフェース(user interface)ともいう。
(3)入出力のしやすさを評価する視点。人間の行動特性を考慮し、身体的、心理的に抵抗感の少ない入出力方式をくふうする。例えば、動作状況、処理進度、終了予想時間の表示、重要度の高いデータの先行出力、マルチメディアデータのインタレース、データのレプリケーションなどはヒューマンインタフェースの改善を目的に開発された手法である。

ヒ

表[table/list]

広義には、互いに関係のある項目が、その相対的位置によって識別できるようにデータを配列したもの。

①文書処理では、データや情報の関係や意味を整理し、わかりやすく表現した書面。文字、数字、記号を主体とするもので、図的要素により視覚化するグラフと区別する。一般的には、縦横のマス目を利用する。(1)一覧表、(2)分類表、(3)比較表(対照表)、(4)評価表などがある。
②プログラムでは、連続する同じ形式のデータ項目にひとつの名前を与えたもの。それぞれのデータは、データ名と要素の番号(添字)で指定する。(類)テーブル、配列

表意コード[mnemonic code]

①人間の言語を連想しやすく規定したコード。アセンブラの別の呼び方。コンピュータが実行できるプログラムは2進数の列で構成するが、直接 2進数でプログラムを組むのは不便である。そこで、コンピュータの各命令を適当な覚えやすい略号にして人間が扱いやすくしたアセンブラ言語ができた。(類)アセンブラ
②データのコード化の一手法。コード化対象の属性を連想しやすい文字や数字で表現するコード。略語コード、子音コードなどが代表的である。
(例)日本→JPN、米国→USA、20インチカラーテレビ→TVC-20

表意定数[figurative constant]

予約語のうち、特定の値を割り当てた定数。データ部で定義しなくても使用できる。単数形と複数形があるが、同じ意味であり、どちらを使用してもよい。(例)ZEROは数値のゼロ、SPACEは文字の空白

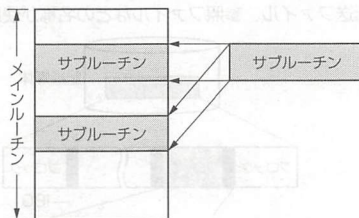
表計算ソフト[spreadsheet program／worksheet program／table oriented database]

定義した表にデータ作成をすると同時に、式や関数を入力して縦横の集計処理などを簡便に行うことができる計算ソフトウェア。

表は縦横に分割してあり、縦方向を行(row)、横方向を列(column)、それぞれが交差している部分を項目欄またはセル(cell)という。文字や数値などのデータは、画面に表示されたセル単位で入力する。2行4列目のセルをR2C4などと指定する。行を数字、列をA、B、C、……で示し、C2などとセルを指定するソフトウェアもある。

表計算ソフトが手書きの集計表と違うのは、このセルの中に数値と数式の両方を入れられることである。しかも数式を入力した場合は、セルにはその結果を表示する。スプレッドシートの機能を拡張して、表の大きさを自由に変える機能、再計算機能、自動実行機能(マクロ機能)、データベース機能、グラフ作成機能、画面分割などの機能がある。

開いたサブルーチン[open subroutine]

同一性格の処理内容のルーチン(サブルーチン)をプログラム中の必要な箇所に挿入できるようにしたサブルーチン。一般的にはソースプログラムライブラリに登録しておき、コンパイル、あるいはアセンブルの際に自動的にソースプログラムの中に組み入れる。アセンブラのマクロルーチン、COBOLのCOPYルーチンなどが代表的。

(関)サブルーチン、(対)閉じたサブルーチン

フ

ファイル[file]

①外部記憶装置上に作成するデータやプログラムの集合。ファイルは論理的な概念であり、ファイルが作成される物理装置の性質を反映する。

②プログラムで処理するレコードの集合。レコードは関連あるデータの並びであり、ユーザが作成、更新する。

【ファイルの分類と名称】

ファイルは、①の視点からは、記録媒体、編成法、アクセス方法などで分類できる。

(1)記録形式による分類：バイナリファイル、テキストファイル

(2)編成法による分類：順次編成ファイル、索引順次編成ファイル、直接編成ファイル、区分編成ファイル、VSAM編成ファイルなど

(3)ボリュームによる分類：単一ボリュームファイル、マルチボリュームファイルなど

(4)アクセス法による分類：直接アクセスファイル、順次アクセスファイル

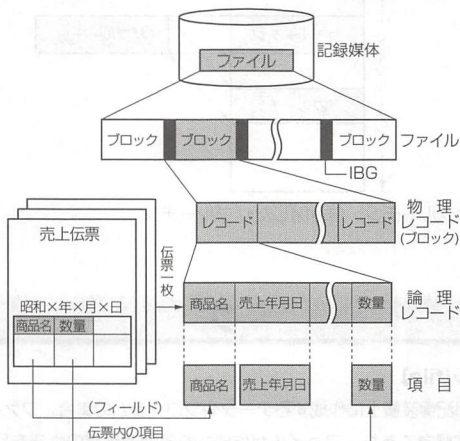
②の視点では、記録した内容や使用目的で分類でき、多くの種類名称がある。

(1)用途による分類：マスタファイル、トランザクションファイル、ワークファイル、経歴ファイル、アーカイブファイル、バックアップファイルなど

(2)使用期間による分類：パーマネントファイル、テンポラリファイルなど

(3)記録内容による分類：文書ファイル、データファイル、画像ファイル、音声データファイルなど

(4)利用主体による分類：システムファイル、ユーザファイル、共有ファイル、個人ファイルなど

この他、ファイル処理の場では、入力ファイル、出力ファイル、中間ファイル、転送ファイル、参照ファイルなどの名称が使用される。

ファイル管理[file management／library management]

プログラムやデータを、ファイルという形式で集中的、体系的に管理す

ること。また、これらをプログラムのライブラリ、ファイルのライブラリといい、これらの管理機能全般をさす。ライブラリは保守や再利用のために保存されるもので、適切に管理されなくてはならない。次の5つの視点で行う。

(1) **スペース管理**：記憶媒体の有効利用、使用頻度に応じた配置などの目的で、記憶領域の再編成、再構成を行う。

(2) **共用管理**：複数のユーザ、複数のプログラムが参照したり、更新するファイルの整合性を維持する。

(3) **障害管理**：障害に備え、ファイルのミラーリングやバックアップを行う。

(4) **変更管理(世代管理)**：更新や変更のたびに変更内容を記録、バージョン管理を行う。変更前のファイルも規定した世代分を保存する。

(5) **統計管理**：ファイル管理のための統計を作成する。

ファイル設計[file design]

データの格納や参照に関する方法や構造に関わる設計。データの種類、分量、用途、参照・更新の頻度などを考慮して、ファイルの構成(データベース構造、ファイル編成など)、様式を決定する。ファイルレイアウトは、データをファイルに格納する際の、項目、桁数、属性などを決定する。

ファイル編成[file organization]

磁気テープ記憶装置、磁気ディスク記憶装置などの外部記憶装置の機能とアクセス方法を適合させるファイルの論理的な構成、編成法の総称。具体的には下記のような編成、VSAM編成がある。

編成法	外部記憶装置の種類	アクセス法
順次編成	MT·DASD	順次
直接編成	DASD	順次·直接
索引順次編成	DASD	順次·直接
区分編成	DASD	順次(直接)
相対編成	DASD	順次·直接

ファイルレイアウト[file layout]　(類)ファイル設計

ファクトデータベース[fact database／source database]

検索の目的となるソース情報を提供するデータベース。コンピュータの処理スピードや記憶容量などの限界が解消するにつれ、ファクトデータベースが誕生した。リファレンスデータベースでは情報の所在を示すだけであるが、情報そのものの提供ができるようになったことにより、商用データベースサービスにファクトデータベースが増加した。データベース利用者の大半がファクトデータベースを対象にしている。

ファクトリオートメーション[factory automation：FA]
製造工場で、製品開発から製造工程まで情報化と自動化を進め生産性と柔軟性の向上を図ること。CAD／CAMにより製品の設計段階からコンピュータを使い、製造工程でのオートメーション化まで含む。工場では、多くの設備、装置、機械、材料を利用しており、連携、統合化の技術はさまざまである。製造段階ではNC工作機械、産業用ロボットなどにより、部品の運搬から組立、検査、梱包、出荷に至る全工程での自動化、省力化が進み、多品種少量生産などの社会的ニーズに対応できる生産体制づくりを目指した統合的オートメーション化が進んでいる。

フィールド[field]
ファイル処理では、1件のレコードを構成する要素で、レコードのもつ各データの格納領域のこと。表計算ソフトでは縦に並ぶ領域であり、レコードの構成によっては1レコード中に複数個のフィールドがある。アイテム(項目)ともいう。例えば、社員レコードの中の社員番号欄、社員名欄、住所欄などの欄をさす。(同)項目

フィールドテスト[field test] (類)運用テスト、(参)ベータ版

フェッチサイクル[instruction fetch cycle／fetch cycle]
コンピュータのCPUは主記憶装置内にある命令を命令レジスタにロードし、解読し、実行することを繰り返す。このとき命令を記憶装置から取り出し始めてから取り出し終わるまでの段階。命令読取り段階ともいう。
(参)命令サイクル

フォアグラウンド処理[foreground processing]
複数のプログラムを同時並行的に処理しているマルチタスクシステムで、バックグラウンド処理に先行する優先順位の高い処理のこと。
(対)バックグラウンド処理

フォーマット[format] (同)初期化

フォルダ[folder]
GUI環境の階層的ファイルシステム(HFS)のディレクトリに相当する概念。名前をもった書類フォルダのアイコン1個で示すファイルの集合。サブディレクトリと同様、フォルダの中にフォルダを入れることも可能である。
フォルダをクリックするとその中身をウィンドウに表示する。フォルダにファイルを入れる操作は、ファイルのアイコンをドラッグして、フォルダのアイコンに重ねればよい。(類)ディレクトリ

フォルトトレランス[fault tolerance]
システムの信頼性を高めるための機能。システムに冗長性を導入して実現することが一般的である。耐故障技術ともいう。ハードウェアの障害に対処するには余分な回路・部品・装置を付け加える。ソフトウェア障害に

は、プログラム誤りの収集分析、故障診断プログラムの利用、二重処理、照合などを行う。耐障害機能を高め、常時稼働能力をもつシステムを、特にノンストップコンピュータという。(参)デュアル構成システム

フォント[font]

書体。画面やプリンタなどに出力する文字の形の情報。文字コードとは別のもので、目に見える文字の書体をさす。同じ文字でもゴシック体や明朝体など複数のフォントが存在する。

深さ優先探索[depth first search]

木構造のグラフのデータ探索法の一つで、ルートからスタートし、分岐点では常にひとつの経路を選択しながら、次第に深い頂点へとたどる探索法。葉に至れば、経路を逆に引き返し前の頂点に戻り、そこからまだ探索していない経路があれば探索を続行する。探索に成功したときは、経路が記憶されている。失敗したときにはルートに戻ってしまう。

(対)幅優先探索

負荷テスト[load test]

一般にシステム開発時のテストは、少量のデータ、回線、端末で実施されることが普通で、本稼働の際、多数の回線や端末から大量のデータが投入された場合のテストがない。この点を特にテストすることを負荷テスト、または過負荷テスト(設計目標以上の負荷をかける)という。端末がいっせいに処理を要求したとき、レスポンスが設計の目標値を満たすかなどを検査する。

不稼働率[non-availability] (対)稼働率

副プログラム[subprogram]

広義には、モジュール化したプログラムで、上位のモジュールから見たときの下位のモジュール(submodule)をさす語。例えば、メインプログラムから呼び出して利用するプログラム。代表的なものにサブルーチン副プログラムがある。このほか関数副プログラム、初期値設定副プログラムなどがある。(参)サブルーチン副プログラム

復帰[return] (関)戻り番地、戻り値、飛越し

プッシュダウン[push down]

スタックに新しいデータを記憶させること。単にプッシュ(押し込み)ともいう。(参)スタック

物理アドレス[physical address] (同)実アドレス

物理データ設計[physical file design]

論理データ設計(ファイル設計)で定義されたファイルのデータ項目、データ量や処理内容から、ファイルの編成方式やレコードレイアウトを定義する工程。記憶装置や記憶媒体を意識したファイル設計で、データベー

ス構造もこの時点で定義される。通常、次のような順で行う。

(1)マスタファイルと関連ファイルのデータ項目、編成法の決定

- ファイルの特性、管理される項目の把握、他ファイルとの関連
- ファイル構造と編成法の決定
- 記憶媒体の決定、容量、入出力時間の算出

(2)レコードレイアウトの設計

- 欄、項目、フィールド、レコード、ファイルの属性、サイズ
- 固定長レコードか可変長レコードか
- レコードレイアウトの決定

物理レコード[physical record]　(関)レコード

不定長レコード[undefined length record]

(関)レコードタイプ

フ

部品化[divide/division]

部品とは、標準化されたソフトウェアを再利用する際の再利用の単位のこと。従来の共通モジュールの考えを拡げたもので、再利用を前提にした標準化により、他のソフトウェアの構成部品になるようにモジュールを作る。これらを整理、分類のうえ部品ライブラリの形で保管しておき、必要に応じて再利用するものである。部品モジュールの数は大量になるため、登録、検索、結合などの作業を支援するツールの活用が不可欠である。

部分順序木[partially ordered tree]

2分木のひとつ。親(ルートに近い頂点)のデータが子(親から出た枝の先の頂点)のデータより小さくならない木のこと。2分探索木が枝の左右でデータの大小を割り振ったのに対比して、部分順序木は枝の高さ(深さ)で大小を割り振る。ソートには適さないが最大値探索に適している。優先度の順位のある待ち行列の実現などに利用する。

プライオリティ[priority]　(同)優先順位

プライバシ保護[privacy protection]

コンピュータセキュリティに関連して、個人に属する情報や個人に関する情報の利用・流出を制限し、個人の権利(プライバシ)を守ること。積極的には、自分に関する情報は自分で管理できる権利を含む。

例えば、銀行取引、クレジット利用状況、病歴、犯罪歴など、個人の人権、信用、秘密にかかわる情報を調査機関などが勝手に収集できないようにアクセスを制限すること。また自分に関する情報の開示を求め、訂正できるようにすることで情報の悪用を防止し、人権を保護する。アメリカでは1980年にプライバシ保護法が成立した。日本では1988年に個人情報保護法を公布した。

プライム領域[prime data area]　(関)索引順次編成

ブラウザ[browser]

データベース、記憶媒体、ファイルの内容を検索したり、表示させたりするソフトウェアの総称。内容を編集したり書き換えたりする機能まではないものをさす。ブラウズ(browse)は、本をパラパラとめくることを意味する。例えば、次のようなソフトウェアがある。

(1) 記憶媒体中のファイルやデータベースのディレクトリ、目次などを表示し、ユーザの選択によりその内容を表示するソフト、そのダイアログボックスなど。主としてプログラム開発に用いる部品の検索やデータ媒体中のファイル検索に用いる。FileManagerやFinderなどもブラウザの一種である。

(2) ネットワーク上の情報提供者の画面を表示するソフト。特にHTMLで書かれたWWWの情報を閲覧するためのクライアントプログラム。ブラウザが示す画面内のアイコンを操作してリンクされたページを見ることも前のページへ戻ることもできる。閲覧ソフト、ビューワともいう。NCSAが開発した最初のグラフィカルなブラウザMosaicを皮切りに急速に普及してWWW人気が定着した。現在はNetscape NavigatorとMicrosoft Internet Explorerが事実上の標準になっている。

(3) データベースの内容を検索したり、一覧したり、表示したりする対話型のソフトウェア。編集モードでは内容を書き換えることもできる。

(4) 開発ツールの一種で、既存のプログラムの所在を調べたり変更するファイル操作プログラム。オブジェクト指向言語に多い。

フラグ[flag]

ハードウェアやデータなどが特定の状態にあるかどうかを表すビット、バイト、キャラクタ、語。標、標識、目印、マーク、旗などともいう。分岐命令が飛越し先を判断するために評価する。

フラグの役割を果たす変数はプログラム上で用意する場合と、条件コードレジスタのようにハード的に用意する場合がある。フラグが2値のいずれかしかとらない場合、特にスイッチということもある。

フラグメンテーション[fragmentation]

もとは半端な部分、断片ができること。情報処理では、主記憶空間に複数のプログラムを割り付けたとき、サイズが合わないための未使用部分(フラグメント)が生じることをさす。

①多重プログラミングを行うためには、主記憶領域を複数の区画に分割して、複数のプログラムを記憶させる。実記憶領域の分割方式には、領域を複数個の固定長の区画に分割し、各ジョブを大きさに応じて適切な区

画の中で実行させる固定区画方式と、主記憶の各区画をジョブの開始時に、ジョブの大きさに合わせて決定する可変区画方式がある。実記憶領域の使用効率は可変区画方式のほうがよい。しかし、多数個のジョブの開始、終了にともない、数多くのこま切れ的な不使用領域が発生する。この不使用領域の発生をフラグメンテーションという。フラグメンテーションの問題を解決するためには、定期的に使用中の区画をひとつの連続領域に集め、こま切れの未使用領域をひとつの連続領域とする。または仮想記憶方式にすれば、大幅に改善できる。

②ファイル装置上の空間を区分しておき、必要な大きさだけファイル空間を割り当てる方式(例えば区分編成ファイル)では、ファイルの更新やレコードの削除により、空き空間(無効領域)ができる。この空き空間をさす。

フ

フラグレジスタ[flag register] (同)条件コードレジスタ

ブラックボックステスト[black box test]

プログラムのテスト法のひとつで、テストデータの選定・作成する際に、プログラムの内部構造・制御の流れにかかわらず、その外部仕様に基づいてデータを選ぶ方式。つまり入力される可能性のあるさまざまな組合せのデータを想定し、プログラムのすべての機能をテストするねらいをもっている。機能テスト法ともいう。結合テスト以降の段階で行うことが多い。次のような形のテストデータを用いる。

- **同値分割**:テスト対象の入力データが取りうる値の範囲の中から、同じ意味をもつ範囲を1クラスとし、分割された各クラスを代表する値をテストデータとして選択する方法。
- **限界値分析**:入力データと出力データを同値クラスに分割し、それぞれのクラスの端(境界)の値をテストデータとして選択する方法。
- **因果グラフ(原因-結果グラフ)**:入力と出力の関係をグラフに表現し、デシジョンテーブルに展開して、テスト項目を設計する方法。

(対)ホワイトボックステスト

プラットフォーム[platform]

もともとは人または物を載せる台の意味。工学では、あるシステムを搭載するための別のシステム。情報処理では、あるソフトウェアやコンピュータシステムを動作させるための基盤となるハードウェアおよびソフトウェア環境、端的には動作環境をさす。

フリーターム[free term]

データベース検索の際に用いる語で、タイトル、コンテンツにその語が存在すればすべて検索の対象とする語。文献や論文のデータベースでは登録時にその内容を表すいくつかのキーワードを登録してあるため、

キーワードの組合せで効率よく目的の対象を検索できる。しかし、適当なキーワードがない場合などキーワードに代えて任意の自然語(フリーターム)で検索でき、主題(タイトル)と本文(コンテンツ・テキスト)を検索する。

プリコンパイラ[pre-compiler]

ソースプログラムのコンパイル前に用いる言語処理プログラム。あるプログラム言語の仕様を拡張し、その拡張した表現を用いて書いたソースプログラムを標準的なソースプログラムの形式に変換する処理を行う。

(類)プリプロセッサ

プリプロセッサ[pre-processor]

①広義には、特定のプログラムのため、入力や処理をしやすくするような形にデータ形式や計算手順を前もって変換する機能をもつプログラム。トランスレータの一種。

②現実的には、コンパイル前のプリプロセス処理(前処理)を行うプログラム。前処理命令の解釈、ファイルのinclude、文字や文字列の置換、マクロ展開等の処理を行う。出力はコンパイルの単位(翻訳単位)となる。

(類)プリコンパイラ

ブリンク[blink]

光などのまたたき、明滅。ディスプレイ画面上のカーソルや文字、図形などを、関心をひくために意図的、周期的に変化させること。

プリントサーバ[print server]

クライアントサーバシステムの中で、プリンタを保持し、ネットワーク利用者(クライアント)からのプリント要求を専門に処理するコンピュータ。大容量の磁気ディスクと高性能のプリンタを装備し、スプール機能を使って効率的に印刷する。

プルダウンメニュー[pull down menu]

サブメニュー表示の一形式。画面の上部に表示したメインメニューバー(処理やファイルの一覧)から主処理の項目を選んだとき、その項目の下に垂れ下がるように表示するサブメニュー。

(対)ポップアップメニュー、サブメニュー

フルテキストデータベース[full text database]

論文や新聞記事などの全文データベース。文献・論文データベースに登録してあるデータを特定する表題や見出し、著者名、収録ライブラリ、要約、キーワードなどの部分をタイトルといい、対象となる論文や記事の本体をコンテンツまたはテキストという。

ブレイクポイント[break point]

①コントロールブレイク処理で、処理手順を変えるための区切り点。例え

フ

ば、在庫の商品別集計の、商品コードの変わり目など。
(関)コントロールブレイク、グループトータル

②プログラムのデバッグや実行状態の監視のため、プログラムの実行を中断し、外部から割込みがかけられるように、プログラム内に指定した場所。チェックポイント(check point)、区切り点ともいう。プログラム内にブレイクポイントを設定することで、プログラムの実行過程を追うこと、実行中のある場所における記憶領域のダンプをとること、デバッガの呼出しを行うことなどの処理を、ハードウェアの助けを借りずに行うことができる。(関)デバッグ

プレプリント[pre-print]

コンピュータの出力用紙に、あらかじめ指定の位置に出力項目名を印刷しておくこと。出力時にはデータのみを印字すればよく、処理の効率がある。(対)オーバレイプリント

フローチャート[flowchart/flowdiagram]

問題の定義、分析または解法の図的表現。流れ図ともいう。演算データ、流れ、装置などを表現するための流れ図記号を用いて表した図表。

流れ図を用いることによって作業やプログラムの処理の流れ、使用する資源が正確に把握でき、第三者にも正確に伝えることが可能となる。流れ図は、資源、オペレーション、判断、コネクタなど、一定の形式、順序、要素にしたがって書く。なお、コーディングは、通常、流れ図に基づいて行われる。流れ図を作成するときに用いる記号群(flowchart symbol)は、JISが定めている。代表的な流れ図には次のようなものがある。

(1) **プログラムフローチャート**:コーディングの指針を与えるためにプログラムの処理手順を概括的に図式化したもの。

(2) **システムフローチャート**:データが発生してから処理を行って情報を得るまでに、システム全体のデータの流れと資源を図式化したもの。システムに関わる部門と役割、ファイルの作成、保存の過程とその記録媒体、伝票や報告書との関わりなどを明確にする。

(3) **データフローチャート**:データを処理して情報を得るまでに必要なデータの流れとその処理名、記録媒体、利用するハードと操作などの関係を明確にする。

(4) **プロセスフローチャート**:作業工程や加工工程の順序を、流れ図記号を用いて見やすくした図表。

プログラミング[programming]

広義には、プログラムを設計し、書き、テストし、運用できるプログラムを作ること。プログラム開発に同義。狭義には、プログラム言語によ

り、プログラムをコーディングすること。
(類)プログラム開発、コーディング

プログラム[program]
　コンピュータに処理させる作業を指令するための手順をプログラム言語により記述した文。一連の命令や文の集まり。その内容や処理の目的によりさまざまに分類できる。狭義にはユーザがプログラム言語を用いて記述するアプリケーションプログラムをさす。広義にはオペレーティングシステムやハードウェア組込みのシステムプログラムを含む。

プログラム開発[program development]
　広義には、システム開発工程におけるプログラム設計からテストまでの広い段階。狭義には、プログラム設計の仕様書に基づいて、データ構造とプログラムの制御構造を決め、各種プログラム言語によりプログラムを記述し、デバッグ、コンパイル、テストなどを経て運用可能なプログラムを作る工程。プログラミングともいう。プログラム開発の効率化をめざし、デバッグツール、CASEツールなどの開発支援ツールの利用も進んでいる。

プログラムカウンタ[program counter : PC]
(同)命令アドレスレジスタ

プログラム言語[programming language]
　コンピュータのプログラムを作成するために用いる人工的な言語。プログラミング言語ともいう。自然言語と異なり、機械的な処理の対象とするため、解釈に依存する部分がないように文や文法(構文や意味)を明確に定義してある。機械語、Fortran、COBOL、ALGOL、BASIC、Cをはじめ、その数はきわめて多く、さまざまな視点で分類できる。

<table>
<tr><td colspan="6">プログラム言語</td></tr>
<tr><td colspan="4">汎用プログラム言語</td><td colspan="2" rowspan="2">特殊問題向け言語</td></tr>
<tr><td colspan="2">計算機向き言語</td><td colspan="2">問題機向き言語</td></tr>
<tr><td>機械語</td><td>アセンブラ言語</td><td>手続き向き言語</td><td>非手続き向き言語</td><td>シミュレーション言語</td><td>リスト処理言語</td></tr>
</table>

プログラムジェネレータ[program generator]
(同)ジェネレータ

プログラム設計[program design]
　システム設計(内部設計)の仕様に基づいてプログラムを開発するための設計段階。プログラム言語、機能別のモジュール分割、プログラム間のインタフェースなどを定め、プログラム仕様書としてまとめる。

プロシージャ[procedure]　(同)手続き

プロセス[process]

①処理の単位を意味する用語。計算(computing)、アセンブル、コンパイル、編集、更新などの意味を広く含む。(類)ジョブ

②利用者のジョブやシステム管理の仕事をコンピュータ内部で実行する実体。オペレーティングシステムがジョブに応じてプロセスを生成し、資源を割当て、実行する。(類)タスク

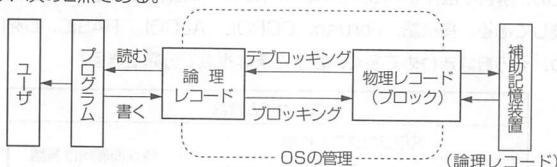
③プログラム実行時に作成される処理の単位。プロセスの中からまた別のプロセスを起動することもできる。このとき、起動したほうのプロセスを親プロセス、起動されたほうのプロセスを子プロセスという。(類)スロット

プロセス制御[process control]

①オペレーティングシステムではタスク制御と同義。プロセスの生成/消滅、プロセス間のシグナルの発信/受信等のプロセス制御を行う機能のこと。

②工業プロセスで、温度、圧力、流量などの物理量をリアルタイムで制御すること。センサなどを通じ装置を監視し、コンピュータでシステムを制御するには高度な処理能力が必要。

ブロッキング[blocking]

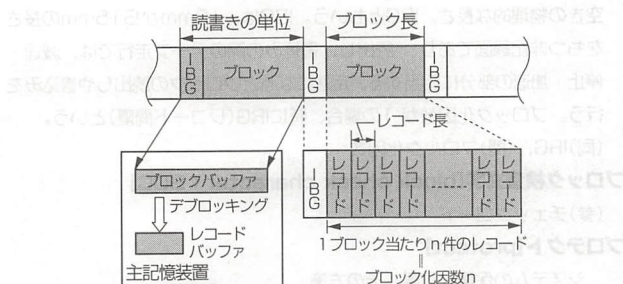
複数の論理レコードをひとつの物理レコードにまとめること。ブロック化ともいう。ブロッキングの指示はJCL(ジョブ制御言語)で行うため、ユーザプログラムの中で特に意識する必要はない。ブロッキングの目的は、次の2点である。

(1)中央処理装置と外部記憶装置の間でファイルの読み書きを行う場合、入出力回数を減らすことで処理時間を短縮する。

(2)IBGを削減して記憶領域を有効に利用する。

(対)デブロッキング、(参)論理レコード、物理レコード

ブロック[block/physical records]

一般にひとかたまりのデータまたはプログラム。

①磁気記憶媒体に情報を記憶させる場合、読み出す場合の、転送単位となる一連の文字や語の集まり。1物理レコードと同じ。ブロックの長さ(文字数、バイト数などで表す)をブロック長(block length:BL)という。文字数、バイト数などで表す。

(類)物理レコード、(参)論理レコード

②構造化されたプログラム中でひとまとまりの処理をする範囲。プログラムの中で共通の名前(変数名など)を使用できる範囲。サブルーチンや副プログラムが、メインルーチンやメインプログラムの指定した名前とは別の名前を利用できるとき、それぞれの名前の有効範囲がブロックである。

Cでは、複数の文を{ }で囲んだもので、プログラムの構成要素のひとつ。このブロックの内部で変数の宣言やデータの初期化を行うことができる。内部変数の有効範囲はこのブロック内部になる。ブロックは入れ子にできる。

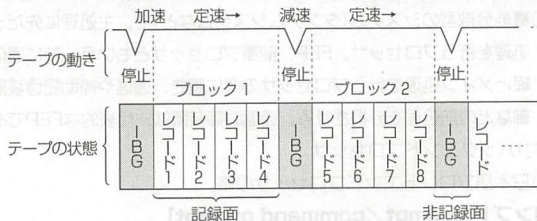
ブロック化[blocking]　(同)ブロッキング

ブロック化因数[blocking factor]

1ブロック(1物理レコード)にまとめた論理レコードの数。ブロック化係数ともいう。ブロック化因数は、磁気記憶媒体に記録できるレコード数を決定する要素である。ブロック化因数を大きくすれば、記憶媒体の記録密度の向上や入出力時間の短縮ができるが、内部記憶装置に大容量のバッファを必要とするため限度がある。通常、ブロック化因数の決定は両方のかねあいで行う。

ブロック化係数[blocking factor]　(同)ブロック化因数

ブロック間隔[interblock gap：IBG／block gap]

磁気テープに記憶させたブロックとブロックの間にある何も記録しない

上の例では，固定長レコードとし，ブロック化因数は4である

空きの物理的な長さ。IBGともいう。IBGは、10mmから15mmの長さをもつ非記録面であり、読出し、書込みの際のテープ走行では、減速・停止・加速の部分に相当する。定速になるとブロックの読出しや書込みを行う。ブロック化因数が 1の場合、特にIRG(レコード間隔)という。

(同)IRG、(関)ブロック化因数

ブロック検査文字[block check character：BCC]

(参)チェックコード

プロテクト[protect]

システムの保護、防護。その方策。

(1)コピープロテクト：フロッピーディスク等の記憶媒体からのソフトウェアやデータの無断コピーを防止するために媒体に行う処理。ガードともいう。これを破ることをアンロック(unlock)という。

(2)アクセスプロテクト：書き換えてはならないデータを他のユーザ、プログラムからの更新要求から保護すること。

(3)シークレットプロテクト：機密のデータを不正アクセスから保護すること。アクセスプロテクトのほかデータの暗号化などを施す。これを破ることをブレーク(break)という。

(4)データプロテクト：ハードウェアの障害からデータを保護すること。物理ファイルに対する更新の履歴を記録するジャーナリング(journaling)、物理的にデータを二重に書き込むミラーリング(mirroring)などの方法がある。

プロトタイピング[prototyping／prototyping model]

ソフトウェア開発においてユーザの要求仕様が明らかになった段階で、ある程度のプログラムを作成し、ソフトウェアの利用イメージをユーザに与え、試用してもらった結果からヒューマンインタフェース部分の要求仕様を固めること。またはその目的のために作るソフトウェア。開発初期段階でユーザの要求仕様が固まれば、最終段階にはいってからの仕様変更による再設計が少なくなり、手戻りの工程を抑制できる。

フロントエンドプロセッサ[front end processor：FEP]

①機能分散型のシステム(タンデムシステムなど)で、主処理に先だって前処理を行うプロセッサ。FEP、前置プロセッサともいう。特に通信回線網とメイン処理を行うプロセッサの間に置き、通信や補助記憶装置の制御などの前処理を分担させる。通信制御装置は、代表的なFEPである。

(対)バックエンドプロセッサ

②日本語フロントエンドプロセッサの略。

プロンプト[prompt／command prompt]

システムがオペレータにコマンドやオペランドの入力を受け付ける準備

ができていることを知らせる表示。一般には画面にプロンプトマークを表示する。あるいはカーソル記号の点滅(ブリンク)等により入力待ち状態であることを知らせる。

分割コンパイラ[separate compiler] (参)コンパイラ

分岐[branch] (同)飛越し

分岐命令[branch instruction] (参)基本制御構造

分散型データベースシステム[distributed database system]

データを複数のサーバ上に置いて、全体としてのデータベースを管理するシステム。分散しているデータベース(DB)をネットワークにより結合し、あたかもひとつのデータベースであるかのようにアクセスできるようにする。分散型データベースは機能的に高度で複雑であるが、データが単一のコンピュータに集中しないため、多数のユーザが同時に使っても応答時間の低下が少ないという長所がある。データの配置には次の2つの場合がある。

(1)データを分割して複数のサーバに配置する場合

クライアント(利用者)間で必要とするデータに偏りがある場合、それぞれ別のサーバ上にあればサーバの負荷分散が可能になる。

(2)同じデータを複数のサーバに配置する場合

サーバの一台に障害が発生しても、他のサーバに同一データがあるため障害の影響を極小化できる。さらに、サーバ負荷分散も可能である。ただし、データの同期の維持、データの更新、性能の低下防止などのためのくふうや余分なリソースが必要となる。

分散データ処理システム[distributed data processing system]

多数の端末やサーバを回線で接続し、互いにデータ処理の役割を分担しながらデータ処理を行うシステム。中央への処理負荷の一極集中を避け、機能を分散して利用現場に密着した処理を行う。代表的なものに、分散型データベース、オンライントランザクション処理、電子メール、電子会議、グループウェアなどがある。次のようなモデルがある。

(1)垂直分散構成(vertical distributed processing system)

ホストコンピュータに接続した複数のサブホストコンピュータ、さらにその下位に接続した多数の端末の間で、業務を分散して処理する方式。例えば、地域に分散する多くの事業所、事業所をまとめる地方支店、それを統括する本店で構成する業種、業務では、各部署の業務は連携していると同時に、独自の処理が必要である。この場合、業務の構成とコンピュータの構成を重ねることで分散処理が可能であるが、その構成は、組織と同様の垂直的な構成になる。

(2)水平分散構成(horizontal distributed processing system)

同規模のコンピュータを多数並べ、それぞれ得意とする機能を分担し

て処理する分散処理方式。コンピュータ間に基本的に上下関係はなく、それぞれ対等である。代表的なシステムにクライアントサーバシステムがある。

(3)ピアツーピアモデル(peer to peer)

IBM社が最初に提唱した、まったく上下関係のない分散処理システムのモデル。ピアは対等な相手の意。すべてのコンピュータが対等な地位にある。専用のサーバを置かないパソコンLANの形態をとったピアツーピアモデルのネットワークOSもある。

文法[grammar]

プログラム言語で、機械語への翻訳を可能にするために表現、構文、構造、意味等を明確化した規定。各言語によりその規則は異なる。

文法エラー[syntax error] (同)シンタックスエラー

分類[sort/sorting] (同)ソート

分類コード[classification code]

コード化の対象データを、大分類、中分類、小分類などの階層構造に分類し、コードの各桁に対応させ、各分類内で一連番号を付けたコードのこと。データが多くなり、階層構造が深くなると桁数も多くなる。産業分類コード、職業分類コード、勘定科目コードなどがある。グループ分類コード、桁別コードともいう。(例)グループ分類コード

分類併合プログラム[sort-merge program]

ユーティリティプログラムのひとつで、ソート(分類)とマージ(併合)の機能をもつプログラム。これらの処理は、サービスプログラムとして提供されるソートプログラムやマージプログラムを使用して行う。入出力ファイルとキー項目の簡単なパラメタ定義をするだけでよい。(関)ソート、マージ

へ

平均故障間隔[mean time between failures:MTBF]

(参)信頼性評価尺度

平均故障継続時間[mean time to repair:MTTR]

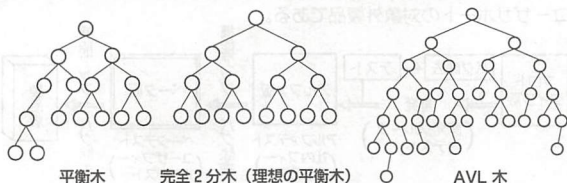
(参)信頼性評価尺度

平均修理時間[mean time to repair:MTTR] (参)信頼性評価尺度

併合[merge/file merge] (同)マージ

平衡木[balanced tree]

2分探索木のひとつ。左右に分かれる枝の先の頂点数がほぼ等しいような形の木構造。はじめはバランスのとれた2分探索木であっても、データ

の挿入や削除が重なるうちに一方に偏ることが避けられない。探索の効率を考えると、右側と左側のデータ件数がほぼ等しいのが理想的で、極端に偏った場合は平衡木に作り直すことを行う。なお常に左右に2本の枝の出る木を完全2分木という。代表的な平衡木にAVL木がある。

並行処理[concurrent processing／overlap processing]

(同)コンカレント処理、(参)マルチプログラミング

並行テスト[parallel test]

既存システムと新システムを同時に同データで並行的に運用して、新システムの機能、性能を検査すること。導入後テスト、運用テストともいう。実データをもとに処理結果が一致するか、障害が発生しないか、テストで見逃したバグがないかなどを検査する。新旧のシステムの性能比較、運用効果の計測なども可能である。

ページャ[pager]

Webサーバのページを作成する支援ツール。Webサーバの各ページ(最上位のページをホームページと呼ぶ)は、HTMLで作成する。HTMLはテキストベースの言語であり、それを見ただけではページのイメージはつかみにくい。そのため、ページのイメージからHTMLを生成するページャ機能が開発された。現在では、ワープロ文書、プレゼンテーションデータ、表計算データなどから、そのイメージをページとするHTML生成機能がそれぞれの作成ツールに存在する。

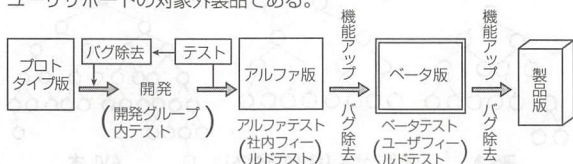
ベースアドレス[base address]

コンピュータプログラムの実行中、アドレス計算において基準として用いる数値。基底アドレス、基準アドレスともいう。ベースアドレスの値は通常、ベースレジスタが保持している。

プログラムが主記憶装置のどの番地に格納されるか不明なため、プログラミング時には最初の番地を0として作成する。実行時には、そのプログラムを割り当てた主記憶領域の先頭の番地がベースアドレスレジスタに記憶され、これにアドレス部の変位を加算したものを実効アドレスとする。

ベータ版[beta version／β version]

完成直前のソフトウェアで、販売前の最終テストの目的で多数のユー

ザに供するバージョン。開発中のソフトウェアで、社内テスト用のアルファバージョンに対比する語。多様な環境や局面で使用してもらうことで、未発見のバグや機能不備の情報を集める。製品版の宣伝を兼ねて、パソコン雑誌の付録CD-ROMやパソコン通信で配布することもあるが、ユーザサポートの対象外製品である。

ヘッダラベル[header label]　(関)ラベル

　データやレコードの前に付加した情報(区切りや属性など)。外部記憶媒体上の、ファイルの先頭に付けたヘッダのこと。
(参)ラベル

ヘルツ[Hertz : Hz]

　周波数、振動数の単位。記号はHz。1秒間に1回振動する振動数(周波数)が1Hzである。電気通信の搬送波やマイクロプロセッサが動作するクロック周波数の表示に用いる。交流の周波数にはサイクル(c/s)という単位も慣用で使う場合があるが、国際単位としてはこのヘルツを使用する。ドイツの物理学者 G.L.Hertzの名前にちなんだもの。

ヘルプ機能[help facility]

　コンピュータシステムの操作性を高めるために、システムの操作法・エラー対処法などのサポート用ドキュメントやガイダンスを、ユーザの求めに応じて画面に出力する機能。ユーザはマニュアルを調べる手間が省け、仕事の能率を上げることができる。

　Windowsのシステムやアプリケーションでは、ヘルプのメニューやアイコンをクリックすると、システムの知識や操作法を教える[目次]、[キーワード]などの選択肢が与えられる。また、ヘルプメッセージを吹き出し図形の内部に表示するものを特にバルーン(balloon)ヘルプ、チップ(chip)ヘルプなどという。

変数[variable]

　高水準言語のプログラムで使用できる基本的な単位。記憶場所に対応する識別子で、規格内の任意の値をもちうる可変因子である。固有の名前(変数名)によって識別し、型によって文字、数値、論理値などの種類が決まる。それぞれを文字列変数、数値変数、論理変数という。

　変数にはプログラム中の文、命令で値を与え、また値を引用することもできる。引用や参照ができる範囲により、大域変数、局所変数に分かれ

る。また、プログラム中で変数の役割により、制御変数、添字、引数、パラメタ、フラグカウンタなどと呼ばれることがある。

ベンチマークテスト[benchmark test]

異なったコンピュータシステムのハードウェア、ソフトウェアの処理能力を比較、評価するために行う試験。システムの処理能力に対するユーザの要求が異なるため、客観的な尺度は存在しない。通常いくつかの課題が決っており、最もポピュラーな機種を基準にして、要求を意識した各種のテスト用プログラムの処理に要した時間などで能力を比較する。

ホ

ポインタ[pointer]

他のデータを参照するためのデータ。

①他のデータの存在場所を示すアドレス。アドレスポインタともいう。または、これを保持するアドレス部やレジスタ、オブジェクト。
(参)スタック

②データベースなどで使用する技法。レコードとレコードの結び付きを表す印。レコードの特定の項目のうち、次のレコードの記憶位置を示す。ポインタによってレコード間の親子関係を結んでいる。

③リスト処理向け言語LISPでは、記憶部分を連続したものとみなさず、小さな記憶単位であるリスト片としてとらえ、各リスト片を結び付けるためにポインタという継手を使う。ポインタには、次のデータを記憶しているアドレスを記憶している。ポインタによって結び付けることにより、単純に一列にならべたり、木構造のようにつないだり、網目状につなぐことができる。(参)リスト処理

④Cでは、他のオブジェクトのアドレスを保持しているオブジェクトのこと。アドレスは、ターゲットマシンのアドレッシングに依存するが、ポインタを使用することによって機種に依存した記述をしなくてすむ。ポインタのさすオブジェクトはどのような型でも構わないが、ポインタ自身はポインタ型となる。

ポインティングデバイス[pointing device]

ディスプレイ画面上の位置座標を入力するデバイス。例えば、マウス、ライトペン、ジョイスティック、トラックボール、タッチスクリーン。

報告書設計[report design]

人間の読める形式に、データをファイルから出力媒体に変換する際の事

項を設計する工程。出力項目と媒体を決定する。報告書の利用目的、印字装置の制約などを考え、用紙サイズ、レイアウト、プレプリントの有無、フォントなどをくふうする。

ホームアドレス[home address]　(参)シノニム

ホームページ[home page]

インターネットのWWWサイトに接続したときに最初に表示される画面。運営する主体が提供するガイダンスページが多く、続けて利用できるページの概要や目次、索引などを記載する。ホームページから表示される画面は単にページと呼ぶ(厳密に区別しないで、すべてのページをホームページと呼ぶ傾向にある)。ページを順序よく選択することで、別の数多くのページにアクセスできる。ページは文字だけではなく画像や音声を扱うことができる。現在はより対話的な内容にするためCGIやJavaのアプレットのような技術によって拡張がなされている。

ホ

ホームディレクトリ[home directory]

ユーザが自分のファイルを置ける最上位のディレクトリ。シングルユーザのシステムではルートディレクトリであり、マルチユーザのシステムでは、各ユーザに割り当てられたサブディレクトリになる。(類)ルートディレクトリ

ホームレコード[home record]　(参)シノニム

保守[maintenance]

①広義には、開発したシステムの最終工程の作業全般。

②狭義には、ユーザによる運用中のプログラムの欠陥を修復する作業全般をさす。通常は運用中のシステムのトラブルを抑制、回避する作業。保全ということもある。

保守体制[field support system]

保守作業を行いシステムを修理、維持するための組織や方法。保守技術者を専任制にしたり、トラブルの警告機能を設けたり、故障診断プログラムを開発するなど保守技術の改善を図ること。

保守性[serviceability／maintainability]

①コンピュータシステムの総合的な信頼性を評価する際の基準となるRASの要素のひとつ。システムが故障したときの修理のしやすさのこと。平均修理時間(MTTR)などの具体的数値を用いて評価することができ、短い時間であるほど保守性のよいシステムといえる。
(関)RAS、予防保全

②ソフトウェア品質の目やすのひとつで、(1)テストがしやすいこと、(2)プログラムの構造や記述がわかりやすいこと、(3)更新や拡張がしやすいことなどをさす。(関)ソフトウェア品質

補助記憶装置[auxiliary storage]

主記憶装置の記憶容量の不足と揮発性によるデータの消失を補う大容量の記憶装置。中央処理装置が直接にアクセスはできない。補助記憶装置のプログラムやデータはいったん主記憶にロードしてから使用するか、入出力チャネルを介してアクセスする。汎用コンピュータでは、磁気テープ記憶装置、磁気ディスク記憶装置など。パソコンシステムでは、フロッピーディスク装置やハードディスク装置など。主記憶装置に比べ処理速度は高速ではないが、大容量であり、差し当たって必要でないデータやプログラムを記憶する。二次記憶装置(secondary storage)、外部記憶装置ともいう。

ホストコンピュータ[host computer]

オンラインシステムなど複数のコンピュータで構成したシステムの中で全体の処理の中核となるコンピュータ。センタコンピュータ、あるいは単にホストともいう。一般的には大型の汎用コンピュータで、大容量のファイル装置や高速プリンタを接続し、ネットワークで結ばれた他のコンピュータに対し種々のサービスを提供する。いっぽう、通信回線を利用してホストコンピュータに外部からデータを入出力する装置を端末という。

クライアントサーバモデルでは、ネットワーク上のコンピュータをすべてホストと呼ぶことがある。

ポストモーテムダンプ[postmortem dump]

プログラムのテストランの最中に、プログラムが異常終了した時点でメモリの内容や変数の値を出力すること。(関)ダンプ

保全[maintenance]　(類)保守

保全性[integrity／maintainability]　(類)完全性、保守性

保存[save]　(同)セーブ

ホットスタンバイ方式[hot stand-by system]

障害発生から復旧までの時間を短縮する技法。デュプレックス構成システムを使い、主系コンピュータに障害が発生したとき、直ちに待機系コンピュータが対応できるようにすること。(対)コールドスタンバイ方式

ポップアップ[pop up]

スタックが記憶しているデータを取り出すこと。単にポップ(跳ね上げ)ともいう。(参)スタック

ポップアップメニュー[pop up menu]

サブメニュー表示の一形式。画面の下部に表示したメインメニューから主処理を選択した後、その項目の上に重なるように表示するメニュー。(対)プルダウンメニュー

ボトムアップテスト[bottom up test]

トップダウンテストとは逆に、階層構造化したモジュールの、最下位レ

ホ

ベルからテストを始め、順次上位のモジュールと結合してテストする方法。下位モジュールをテストする場合、その上位レベルのモジュールが未完成であるときは、ドライバと呼ぶテスト用プログラムを使用する。

ボトムアッププログラミング[bottom up programming]

トップダウンプログラミングとは逆に、最初に小規模なモジュールを考えて、これらを組み合わせ次第に大きな手順とし、最後に全体を構成する方式。ただし、これはトップダウンプログラミングに対比する概念であって、実際のプログラミングをボトムアップ方式で行うことはほとんどない。

ボトルネック[bottle neck]

①コンピュータシステム全体の資源の中で最も負荷の高い資源。溢路ともいう。システム全体の処理効率を考えるとき、他の資源に遊び時間が生じるため、この部分を緩和できれば全体の効率があげられる。びんの狭い口の部分が中身の自由な出入りを妨げることから、流れの妨げとなる部分をさす。

②システム開発等の際に、その開発の妨げとなる事象。

ボリューム[volume]

磁気テープや磁気ディスクなど、ファイルを格納する媒体を管理する単位。ボリュームにはボリューム名を付け、入出力媒体を特定するために使用する。ボリュームの容量と収録するファイルの容量、本数の関係から、表のような呼び方をする。

ボリューム	単一ファイル	複数ファイル
単 一	シングルファイルボリューム	マルチファイルボリューム
複 数	マルチボリュームファイル	マルチボリュームマルチファイル

ボリュームラベル[volume label] (参)ラベル

ホワイトボックステスト[white box test]

プログラムのテスト法のひとつ。テストデータを選定・作成する際に、プログラムの内部構造・制御の流れに基づいて、すべてのモジュール、ルーチン、文を実行できるように選ぶ方式。つまりプログラムのすべての局面を網羅的にテストするねらいをもっている。構造テスト法ともいう。モジュールテスト(単体テスト)の段階で行うことが多い。以下のようなテストがある。

【データの作成手法】

- **命令網羅**:すべての命令を少なくとも1回は実行するように、テストケースを設計。
- **判定条件網羅**:命令網羅を含み、かつ、判定結果が、真または偽になるすべてのケースを網羅。

ホ

- **複数条件網羅**：プログラム中の条件付き分岐命令が判定するすべての組合せを網羅したテストケースの設計。

翻訳プログラム[translator]

①翻訳する計算機プログラムの総称。

翻訳するという表現	翻訳プログラム	翻訳前の言語の例	特　徴
アセンブルする assemble	アセンブラ	アセンブラの言語	アセンブラ言語だけが対象
コンパイルする compile	コンパイラ	COBOL FORTRAN	原始プログラム →目的プログラム
解釈する interpret	インタプリタ	BASIC	1命令づつ翻訳しながら実行
生成する generate	ジェネレータ	RPG	パラメタで機械語を作る

(類)言語処理プログラム

マークアップ言語[markup language]

文書のテキスト属性(文字サイズ、字体、文字修飾、位置など)、体裁(余白、タイトル、目次、見出し、本文、参照関係、図表、索引など)、他の文書やデータの動的な挿入(ハイパーテキスト)、修正箇所の版管理などを論理的に定義できる言語。これらの定義は<　>で囲んだタグを使って定義する。代表的なマークアップ言語としてHTML、SGML、XMLがある。これらによって、大量のマルチメディア文書の提供や交換が容易になる。マークアップ言語で書かれた文書を定義の通り表示、印刷、検索できるプログラムをブラウザと呼ぶ。(関)タグ、ハイパーテキスト、HTML、SGML、XML

マージ[merge]

ソートされた2本以上の列(データ、レコード、ファイル)を合わせて、1本のソートされた列にすること。併合ともいう。

(1) 一定の順序で分類してある(ソートずみの)二つ以上のファイルを、ひとつのファイルに同じルールでまとめること。ファイルのマージという。

(2) 別々に開発した二つ以上のプログラムを全体としてひとつのプログラムにまとめること。プログラムのマージという。

マージソート[merge sort]

マージの技法を応用して行うソートのアルゴリズム。外部ソートに使われることが多い。基本的な考えは、ソートのすんだ複数の列の一端から1件ずつデータを取り出し比較し、最も小さいデータを別に用意した列に移す。記録したデータがあった列の一端から次のデータを取り出し、再び複数のデータを比較し、最も小さいデータを別の列の次に移す。これを繰り返していけば、別の列に全データのソートが完了する。

実際には、外部記憶装置とのデータの読み書きの効率やファイルの終わりを判断する機能を改良して、各種のアルゴリズムが開発されている。

平均計算量は、n件のデータついてnlog2nに比例、最悪の計算量はnの2乗に比例するが、実用上よい

マイクロ[micro]

10^{-6}(100万分の1)を表す補助単位。マイクロ秒は100万分の1秒。

マクロアセンブラ[macro assembler]

プログラム上でよく使う一連の処理(命令群)を、まとめてひとつの命令として定義し、ほかの命令と同様に使えるような機能を追加したアセンブラ。一連の処理をまとめた命令をマクロ命令といい、このような言語をマクロアセンブラ言語という。アセンブルの際には、マクロ命令は機械語の

命令群を組込み、一般の命令同様に機械語コードに変換する。
(関)アセンブラ

マクロ言語[macro language]

マクロ機能の強化、アプリケーション記述との統一化を図ったマクロ命令の記述用言語。アプリケーションプログラムの自動運転のプログラムだけでなく、条件分岐などを含んだ複雑なプログラミングも可能である。マイクロソフト社のVisual Basic Programming System Applications Edition、アップル社のAppleScriptなどは、自社のアプリケーション間で共通に使用できるマクロ機能を提供している。

マクロ命令[macro instruction]

ソースプログラムを簡単に書くために、複数の命令で構成する処理を1命令(パラメタ指定を含んでもよい)で置き換えた命令。

①ソースプログラムの中で、特別な意味をもつ命令で一連の機械語命令と等価な命令。アセンブラのマクロ命令がこれに相当する。
(関)マクロアセンブラ

②原始言語で記述された命令の中で、あらかじめ同じ原始言語で定義した一連の命令に置き換える命令。一連の処理を定義することをマクロ定義といい、マクロ定義したルーチンを翻訳前に挿入することをマクロ展開という。(参)プリプロセッサ、プリコンパイラ

③表計算ソフトやデータベース言語などで、あらかじめ処理手順や処理対象ファイルを登録しておき、起動時の一連の操作に代える指令。
(参)表計算ソフト

マスク[mask]

隠すこと。例えば、数字を表現する文字コードから数値を取り出すために、上位4ビットを除去する必要がある場合、ビット列から不必要な部分を削除するためのビットパターンをマスクという。削除の結果は、原データとマスクの論理積(AND)演算で得られる。

マスタ索引[master index] (関)索引順次編成

マスタファイル[master file]

比較的長期間にわたって保存し、しばしば参照するファイルで、更新しながら継続的に使用するファイル。台帳として使われるファイルに相当する。基本ファイル、永久ファイル、保存ファイルとも呼ぶ。これに対しデータ発生のつど作るファイルをトランザクションファイル、発生ファイル、一時ファイルなどという。(対)トランザクションファイル

待ち時間[latency/waiting time/delay]

ある動作の要求があった瞬間からその動作を開始する瞬間までの時間。
①磁気ディスク装置に対する読み書きの要求をしたとき、アクセスアーム

を所定の場所に位置づけ(位置決め時間)、特定のデータの記録場所が回転してきた後(回転待ち時間)読み書きが始まる。読み書きが始まるまでの時間を待ち時間という。

②スプーリングを行うシステムで、ジョブを投入してから実行するまでの時間。

③オンラインシステムで、トランザクションの要求をしてから応答が返ってくるまでの時間。(参)応答時間

マッチング[matching]　(同)突合せ

マニュアル[manual]

①コンピュータ資源を使うオペレータ、プログラマ、ユーザのための取扱い説明書。手順、命令、操作法などのほか、エラー処理、トラブル処理の対処法なども含む。原義は便覧、案内の意。(類)ユーザズマニュアル

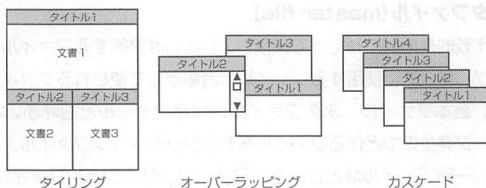
②自動(オートマチック)処理に対比する手動・手作業(マニュアル)。

マルチウィンドウ[multiwindow]

複数のウィンドウ(窓)をディスプレイの画面上に重なり合うように表示する環境。ワープロソフトや表計算ソフトなどで、あたかも机の上で何枚もの書類を広げて作業するように、複数のファイル(プログラムや文書)を画面上に表示し、内容の参照、他の文書の指定部分を別の文書へ挿入することなどができる。複数のウィンドウ表示には次のような手法がある。

(1) **タイリング方式**：画面を上下(または左右)に分割して、双方に異なったウィンドウを表示する。

(2) **オーバラッピング方式**：一般的なマルチウィンドウでは、画面の自由な場所に、自由な大きさで複数の文書を上下に重ね合わせながら表示する。下のウィンドウのどの部分でもクリックすれば、そのウィンドウが最上面になる。

(3) **カスケード方式**：上部のタイトルバーを常に表示する形で、上下、左右に少しずつずらしながら表示すること。カスケードの原義は、庭園などで水を階段状に流す多段滝のこと。

マルチタスク[multitask]

CPU時間を分割し、複数のタスクに割り当てるOSの機能。並行して複数のジョブが生成したタスクにCPUを割り当てることで、マルチプログラ

ミングを実現する。常にひとつのプログラムしか実行できない単一タスク(single task)に対比する語。(参)タスク、バックグラウンド処理

マルチファイルボリューム[multifile volume] (関)ボリューム

マルチプラットフォーム[multi platform]

複数の動作環境上で実行可能なシステムの性質のこと。マルチプラットフォームシステムでは、各動作環境とインタフェースを合わせた中間のソフトウェア(ミドルウェア)をそれぞれ用意する。そのソフトウェアを動作環境に合わせて差し替えることによって、複数の動作環境で動作可能としている。アプリケーションをマルチプラットフォーム化するためのミドルウェアとしては、データベース管理システム、通信管理システム、GUIプログラムなどがある。

マルチプレクサチャネル[multiplexer channel : MXC]

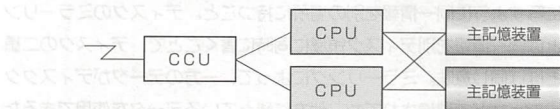
比較的低速なカードリーダやラインプリンタなどの複数の装置に接続する入出力チャネル。多重チャネルともいう。同時に多数の装置を並行的に制御できるような構造から、多数の周辺装置で共用する。データ転送はバイト単位またはブロック単位で行う。バイト単位で行うものをバイトマルチプレクサチャネルという。

マ

マルチプログラミング[multiprogramming]

1台のコンピュータで同時に複数のプログラムを実行させる処理方式のこと。多重プログラミングともいう。入出力装置の速度は、中央処理装置の処理速度にくらべて遅く、入出力操作が行われている間は中央処理装置は待ち状態になる。そこで、プログラムを処理する記憶域(区画)を複数用意して、それぞれの区画に複数のプログラムを記憶させておき、あるプログラムが入出力のために待ち状態となっている間に、他のプログラムを実行するようにした。見かけ上、複数個のプログラムを並行して実行する状態をつくり出すことができ、コンピュータ資源をむだなく、利用できる。

マルチプロセッサシステム[multiprocessor system]

コンピュータの演算能力を拡大するために、複数の主記憶装置を複数のCPUで共通に使い、全体としてひとつの大きな計算機システムとして用いるシステム構成。ひとつのOSが複数のCPUを制御するため、並列処理が可能になり、システムの効率が上がる。またCPUのひとつが故障してもOSが自動的に切り離して処理を続行できるため信頼性が高い。

丸め[rounding]

高精度の数値表現から、低精度の数値表現を得る方法のひとつ。すなわち、数値の下位の桁を調整して意図的に余分な桁を切り落とすこと。調整の方法として、四捨五入、切捨て、切上げなどがある。

丸め誤差[rounding error]

数値演算で有効桁数の考察を誤り、整数演算と実数演算の混用、または単精度、倍精度などの混用、外部表現と内部表現の差異が原因で生ずる誤差。(参)誤差

マンマシンインタフェース[man machine interface：MMI]

(同)ヒューマンインタフェース

ミ

ミドルウェア[middleware]

オペレーティングシステム(OS)とアプリケーションソフトウェアとの中間に位置するソフトウェア群。ハードウェアやOSの違いの差を吸収する役割をになう。ミドルは、中間、中位、仲介などの意。具体的なミドルウェアとしては、DBMS、CASEツールなどのソフトウェア開発支援ツール、GUI制御プログラム、通信ソフトウェアなどをさす。

大規模コンピュータシステムのダウンサイジングやオープン化の流れの中で、ソフトウェア資産の保護や活用のためには、異なるプラットホーム間でのデータの移行や連携が欠かせない。特に、オープンシステムでは、マルチプラットフォーム環境でのシステムの構築が求められており、異機種間にまたがり管理や運用、利用の互換性を確保するミドルウェアの重要性が高まっている。

ミドルウェア
- DBMS
- 通信管理システム(LAN制御含む)
- ソフトウェア開発支援ツール(CASE含む)
- 第四世代言語
- EUCツール
- GUI制御
- ワードプロセッサ
- グラフィック処理
- 運用管理ツール

ミラー／ミラーリング[mirror／mirroring]

①鏡に写すように同一情報を別の場所に持つこと。ディスクのミラーリングとは同一情報を別ディスク領域に同時に書くことで、ディスクの二重化とほぼ同じ意味。ミラーリングによって、一方のデータがディスククラッシュなどで破壊されても、他方に残っているデータを使用できるた

め、信頼性が非常に高くなる。

②インターネットでは、ミラーサイトの略称。

ミリ[milli：m]

10^{-3}(1000分の1)を表す補助単位。ミリ秒(millisecond：msec：ms)は1000分の1秒。

ム

無限ループ[endless loop／permanent loop] (参)ループ

無条件飛越し[unconditional jump] (関)飛越し

無停止コンピュータ[non-stop computer]

システムの信頼性を高度に高め、長期間、また24時間、365日の常時稼働能力をもつシステム。例えば、24時間化している国際経済ネットワーク、電話や発電所、交通信号などを常時制御するシステムは、故障の許されない無停止コンピュータを使用する。耐故障システムは冗長性を導入して実現する。ハードウェアの障害に対しては余分な回路・部品・装置を付け加え、ソフトウェア障害には、プログラム誤りの収集分析、故障診断プログラムの利用、二重処理、照合などを行う。

(参)フォルトトレランス

無停電電源供給システム[uninterruptible power supply system：UPS] (同)UPS

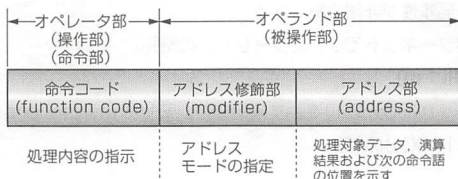
メ

命令[instruction]

コンピュータに演算やその他の処理動作を知らせるコードで、プログラムの基本的な単位。実行すると制御装置や主記憶装置の状態が変化する。

①(機械語命令)コンピュータに演算やその他の処理動作を知らせるコードで、プログラムの基本的な単位。機械語の命令は、記憶装置のデータや入出力機器を区別し、各装置の指定番地に働きかける。

②アセンブラ言語でプログラムを記述する命令コード。その形式はハードウェアに特有で、一般に操作(operation)を指示する命令コード部(操作部、命令部ともいう)と処理の対象となるデータの記憶場所(レジスタや主記憶装置の番地を示すアドレス部)を指示するオペランド(operand)部で構成する。

オペレータ部（操作部）（命令部）	オペランド部（被操作部）	
命令コード (function code)	アドレス修飾部 (modifier)	アドレス部 (address)
処理内容の指示	アドレスモードの指定	処理対象データ，演算結果および次の命令語の位置を示す

③高水準言語でプログラムを記述する命令は、文(statement)といい、オペレータが直接的にシステムに与える操作する命令は、コマンド(command)ということが多い。

命令アドレスレジスタ[instruction address register]

次に実行する命令のアドレスを記憶するレジスタ。プログラムカウンタ、逐次制御計数器ともいう。プログラム内蔵方式のコンピュータは、命令語もデータも区別せず主記憶装置内に格納している。通常1命令語は1ワードか2ワードを占有するので、ひとつの命令の実行のたびに、カウンタに+n(命令が1ワードなら+1、2ワードなら+2)を加算して更新する。命令を実行すると改めてプログラムカウンタの持つ番地を読み取り、順次実行する。ただし、飛越し命令(分岐命令)の場合は、プログラムカウンタの内容を飛越し先のアドレスに設定する。

命令サイクル[instruction cycle]

逐次制御方式の計算機で、次の(1)〜(3)までの計算機の動作段階をさす。

(1)プログラムカウンタが指定した番地の命令を主記憶装置から取り出して命令レジスタに入れる(フェッチサイクル)。

(2)命令コード部を命令デコーダで解読する(実行サイクル)。

(3)命令のアドレス部を演算し、実効アドレスにデータを出す(実行サイクル)。(参)フェッチサイクル、実行サイクル

命令部[instruction field]

コンピュータの命令語の構成要素で、各装置に指示する処理の動作を指示するフィールド。

命令レジスタ[instruction register]

制御装置の一部で、記憶装置から読み出した命令を受けとり、実行するために一時記憶しておくレジスタ。

メインテナンス[maintenance]　(同)保守

メインテナンスフリー[maintenance free]

保守不要のこと、あるいは保守契約のないシステム。コンピュータシステムの維持には保守は必要である。しかしコンピュータの小型化・低価格化が進み、オフィスや一般家庭まで普及するようになった状況では、大量のマシンをメーカが個別に保守することはコストがかかる。そこで価格と

保守費用を下げるため、製品自体を故障がおきにくい設計、仕様にして、保守作業を最大限省けるようにする。

メインフレーム[mainframe]

用途を特定しない汎用コンピュータ。大型から小型のものまで規模はいろいろあるが、通常は運用保守の常駐要員を必要とする規模のクラスをさす。一般的には、ある機関や会社のホストマシンとして事務処理や科学技術計算の中核に使用するコンピュータ。近年は既存のソフトやデータ資産を蓄積、活用するデータベースサーバとして位置づけられているため、ハイエンドサーバと呼ぶこともある。

メインプログラム[main program]

プログラムの中心になる部分。主プログラムともいう。プログラムは、中心となる部分(メインルーチン、制御セグメント、上位モジュール)とそこから呼び出す部分(サブルーチン、従属セグメント、下位モジュール)からなる。小さなプログラムではメインプログラムだけで作成することもあるが、多くのプログラムはメインプログラムとサブプログラムから作る。プログラムの実行にはメインプログラムが必ずなければならない。

メインメモリ[main memory] (同)主記憶装置

メインルーチン[main routine]

(同)メインプログラム、(対)サブルーチン

メガ[mega:M]

10^6(100万)を表す補助単位。情報処理ではメガビットは、2^{20}ビット=1024×1024ビット=1048576ビットのことをさし、メガバイトは1024×1024バイトのことが多い。

メタデータ[metadata]

データそのものではなく、データの意味を記述するデータのこと。データの操作を定義する前提となる。例えば、(6、6、24)はデータそのものであるが、これが(学年、組、学生番号)であるのか(年、月、日)であるのかの意味がメタデータである。通常、データの意味は、それを扱うプログラムやシステムが定義する。DBMSでは、データとメタデータの双方をデータとして扱うので、メタデータのメタデータを定義しておく。これを記録しておくのがシステムカタログ(system catalog)またはDD/Dである。

メニュー[menu]

料理の献立表(メニュー)と同じように、ユーザが選択指定可能なプログラムやファイルを一覧表にして表示した画面。パソコンなどでデータを処理する場合、作業の手順や内容を画面に表示して利用者に選択させる。階層型メニューは、選択項目が多い場合に選択肢に階層構造を持たせたもの

で、メインメニュー、サブメニューなどに階層化してある。また、メニューを常時、画面の上部(または下部)に表示する行をメニューバー(menu bar)という。

メモリ[memory]

コンピュータのデータを記憶させ、CPUが直接に値を参照する記憶装置。一般的には主記憶装置をさし、ハードディスクやフロッピーディスクなどの外部記憶装置とは区別する。(類)主記憶装置、RAM

メモリアドレスレジスタ[memory address register : MAR]

制御装置内のレジスタのひとつ。命令実行段階で主記憶装置に対してアクセスを行うときにそのアドレスを一時的に保持するレジスタ。アクセスの対象が命令語の場合には特に命令アドレスレジスタという。
(参)メモリデータレジスタ

メモリ管理[memory management]

プログラム実行中に動的にメモリを操作すること。コンパイル時の静的なメモリ割当てと対比して動的メモリ割当てとも呼ぶ。

モ

メモリダンプ[memory dump]　(関)ダンプ

メモリデータレジスタ[memory data register : MDR]

制御装置内のレジスタのひとつ。主記憶装置からデータを読み込んだときや逆にデータを書き込むときに、そのデータを一時的に保持しておくレジスタ。記憶データレジスタともいう。(参)メモリアドレスレジスタ

メンバ[member]

①区分編成ファイルを構成する互いに独立したデータ群。区分編成ファイルは直接アクセス記憶装置上に存在し、いくつかの互いに独立したデータ群(子レコード)の部分と、おのおののデータ群に関する情報を持つ登録簿で構成する。メンバ内の編成は順編成ファイルと同じである。
(関)区分編成ファイル

②データベースでは、異なるレコード間のつながりの構造を表すためにひとつの親レコード(owner)と複数の子レコード(member)をセットと呼んで定義する。データベース設計者は、この構造を利用してデータの蓄積、管理、呼出し、結合を行う。

③Cでは、構造体、共用体の個々の要素のこと。(参)構造体

モ

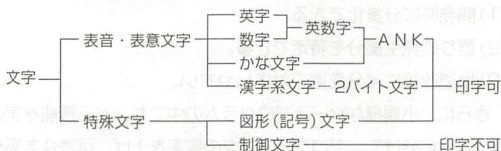
網型データベース[network database]

(同)ネットワーク型データベース

文字演算子[character operator]

プログラム中で文字型のデータを対象に演算を施すときの記号。例えば、二つの文字列を連結する連結演算子がある。Fortranでは、'AB'／／'CD'は文字列'ABCD'になる。

文字集合[character set]

コンピュータシステムで使用できる文字の種類。キャラクタセットともいう。ハードウェアやプログラム言語によっても使用できる文字集合が異なる。なおプログラムを入力するときに使用する文字と、プログラムを実行するときに使用する文字が異なることもある。

(1)Cでは、ソース上に記述される文字集合は基本文字集合といい、英字、数字、記号文字などである。記号文字中の特殊文字は、エスケープシーケンスによって記述することができる。実行時に使われる文字集合はターゲット文字集合といい、ヌル、スペース、改行などの制御文字が含まれる。さらに、ターゲット文字集合は地域化の指定により、国や地域による表記の違いを設定することができる。

(2)JISには、1バイトで表現できる7単位と8単位の情報交換用符号と2バイトで表現する情報交換用漢字符号系がある。漢字コードでは第一水準として特殊記号147文字、数字10種、ローマ字52種、平仮名83文字、カタカナ86文字、ギリシア文字48種、ロシア文字66種、ケイ線素片32文字の合計3489文字、第二水準として3388文字を定めている。

(関)特殊文字、機能キャラクタ

文字定数[non-numeric literal]

文字列で構成する定数。両端を引用符で囲んで表現する。計算の対象にできない。プログラム言語により、引用符、使用できる文字、文字列長の制約、引用符自身の含め方は異なる。

モジュール[module]

原義は、(1)測定の単位・標準、(2)全体の中に組み込む部分または組込みの基準尺度。情報処理の分野では、ひとつ以上の機能を有する部分。構造化プログラミングでは、ある機能を有する小プログラムのこと。

①ハードウェアではメモリや各種の回路を機能単位に分割し、組込みや増設が容易に行えるようにまとめた単位をモジュールという。モジュール

はそれ単体で定めた機能を有し、より大きな単位の部品、付属品の形態になる。

ひとつのモジュールは単位として取り除き、別のモジュールに取り替えることができる。モジュールで構成したハードウェアは、拡張性、融通性、保守性にすぐれた特性をもつ。

②構造化プログラミングでは、ある機能を有する小プログラムをモジュールという。容易に交換し得るような機能単位のプログラムをさす。大きなシステムを構築する場合には、プログラムをモジュール化して作成すれば、次のような利点があり開発効率の向上が期待できる。

(1)開発時に分業化できる。

(2)誤りの発生部分を特定できる。

(3)機能改善に部分変更で対応しやすい。

さらに、小規模なソースプログラムの中でも、ある機能を実行する小部分ごとに分けて、プログラミングの能率を上げ、可読性を高める。構造化プログラミングの普及により、プログラム開発を各機能ごとに部分に分けて行うようになった。

③プログラム開発の段階では、プログラム単位を表す用語で、コンパイル前のソースプログラム(原始プログラム)をソースモジュール、コンパイラの出力プログラム(目的プログラム)をオブジェクトモジュール、リンカの出力プログラム(実行形式プログラム)をロードモジュールという。

④ひとまとまりの手続きやデータの宣言からなる言語の構成要素で、他の同様の構成要素と相互に作用するプログラム単位。副プログラム、サブルーチン、手続き、関数など個別にコンパイルできる範囲をさす。例えば、呼び出すプログラム、呼び出されるプログラムはそれぞれモジュールである。

モジュール化[modularization]

大規模なプログラムを開発する際に、プログラムを各機能ごとにいくつかの小部分に分けて行うこと。分割の方式やサイズは、システムで異なり一定していないが、処理機能やデータ構造から判断して、独立性の高い単位に分割する。

モジュール間インタフェース[interface between modules]

モジュール相互間のインタフェース。構造化設計の段階で分割するモジュールの相互間で授受するデータの定義といえる。すなわち、下位(従属)モジュールが上位モジュールからどんな情報を受け取り、どんな情報を返すかを明確にすることで、モジュール間インタフェース定義図で表現する。

これに対し、プログラム間インタフェース(interface between programs)は、モジュール内のプログラム相互間のインタフェースで、内

部設計の機能分割の段階で行い、一般にデータフローダイヤグラム(DFD)やデータ流れ図で表現する。

モジュール言語[module language]

広義には、特定の機能を実現するための手続き(プロシジャ)群からなり、アプリケーションプログラムから呼び出して、実行すべきモジュールを定義するかたちで使用するタイプのプログラム言語。狭義には、アプリケーションプログラムがデータベースのデータ操作を要求するためのモジュールを定義する言語処理系。適用業務プログラムからのデータの独立性を保証するために、DBMSとのインタフェースを果たすモジュールを設ける。

モジュールは手続きのひとつであり、DMLで記述できる。親言語のプログラムが呼出し文で指定すると、モジュール内で定義されたデータベース操作が実行される。

モジュール構造[module structure]

システムをモジュールに分割して構成するときの、論理的な構造。モジュール分割を階層構造的に行えば、プログラム開発は全体から部分へと段階的にトップダウン展開することで実現できる。また、モジュールの独立性を高くすることも重要であり、これはモジュール間結合度を弱く、モジュール独立性を高くする方向で検討する。

モジュール設計[module design]　(類)モジュール化

モジュールテスト[module test]

プログラムを複数のモジュールで構成するとき、各モジュールの機能単位ごとに独立したテストを行うこと。単体テストともいう。個々のモジュールは小規模で、機能が限られていることが多く、テストケースの設定、テストデータの作成も比較的容易であり、エラーを検出した場合も、原因の特定、修正は比較的容易である。ただし各モジュールが単独で実行可能でない場合は、スタブ、ドライバなどのテスト用プログラムが必要になる。単体テスト後はモジュールの結合テストを行い、プログラム間インタフェースや全体としての機能の一致度を検証する。

(関)結合テスト、総合テスト

文字列[character string]

文字だけから成り立っている列。記号、英字(アルファベット)、2進数字などの列のこと。文字を連ねることで、文字数より多くの情報を表す。また人間にわかりやすい一意名を定義できる。文字列処理とは、文字列の集合から、単語の抽出、特定の語の検索などをさす。

文字列検査[inspect]

データ項目中の文字や文字列の出現回数を数えたり、それらを他の文字や文字列に置き換えたりすることをいう。

文字列検索[word search／word retrieval] (同)文字列探索

文字列操作[string manipulation／operation of character strings]

文字列の分解、連結、置換、比較、文字や文字列の出現回数のカウントなどの文字列について操作機能。

文字列探索[string searching／string pattern matching]

ファイル名や文書中から指定した文字列を探し出す機能。アプリケーションによっては探し出した文字列の先頭にカーソルを移動し、反転表示するなどの機能を付加することもある。検索方法には次のようなものがある。

(1)**部分一致検索**：指定文字列をファイル名の一部にもつファイルの検索。

(2)**あいまい検索**：大文字と小文字、ひらかなとカタカナの区別をせず、記号文字を無視する検索。

(3)**インクリメント検索**：検索文字列を入力し始めた時点で検索を開始、入力に応じて検索範囲を絞る検索。電子国語辞典やキーワード集など見出し語が一定の順に並ぶ場合の検索に適用する。

(関)あいまい検索、インクリメンタルサーチ

戻り値[returned value／return value]

関数や副プログラムに値を引き渡して実行した後、関数や副プログラムから戻される値。返り値、返却値ともいう。

戻り番地[return address]

外部関数や副プログラムに制御を移し、その実行が終了した後、制御が戻るべきアドレス。復帰アドレス、返り番地ともいう。

モニタ[monitor]

①共有資源へのアクセス状況を監視、制御、または検査するプログラムや装置。

②アクセスできる場所を共有資源とし、共有資源に対する操作を含めて相互排除すべき同期を局所化して定義すること。利用者が直接アクセスする部分を限定するOSの構成法。プロセスを制御する同期基本命令。

③ディスプレイ装置の別称。

問題向き言語[problem oriented language：POL]

ある種の問題について解法を記述するために特に適したプログラム言語。ALGOL、Fortranなどの手続き型言語、GPSS、SIMSCRIPTなどのシミュレーション用言語、LISP、IPLVなどのリスト処理用言語、情報検索用言語等の総称。(参)特殊問題向き言語

モ

有効アドレス[effective address] (同)実効アドレス

有効桁[significant digit]

数表示で、与えた目的、正確度、精度を保つために必要な数字。コンピュータの種類により内部の数値表現の精度はいろいろある。

1ワードを32ビットとするコンピュータでは、2の補数による固定小数点表示では〔-2^{31}～$2^{31}-1$〕の範囲の数値しか表すことができない。しかし、科学技術計算などではより広い範囲の数値を扱うため、浮動小数点表示を使用し、高い精度を確保する。単精度実数型の場合は1ワードを 4バイト(32ビット)で表現し、長精度実数型は2ワード、8バイト(64ビット)、さらに高い精度を要求する場合は、倍精度実数型を採用し、4ワード、16バイト(128ビット)を使用して数値を表現し、計算する。

ユーザID[user identifier／user identification]

利用者個人を識別するために与えたコード(IDコード)、またはIDコードにより利用者個人を識別すること。入退室、ファイルアクセス等の際に資格を明らかにし、利用権を登録するためのコード。第三者が知ることができる利用者名と第三者が知ることができないパスワードからなる。利用者名は現在誰が使用中であるかを調べるためにあり、パスワードは本人確認のためにある。IDカードはIDコードのカード状の入力媒体。

ユーザインタフェース[user interface]

(同)ヒューマンインタフェース

ユーザサポート[user support]

ハードウェアやソフトウェアのユーザに対して、操作教育やトラブル対応、情報提供などのサービスを提供し、効果的な運用を支援すること。必ずしもすべてのユーザに均等に無償で行うわけではなく、内容に応じて個別に契約を結ぶのが基本である。(関)サポート

ユーザズマニュアル[user's manual]

コンピュータシステムを使うユーザのための取扱い説明書。操作説明書ともいう。運用規定、手順、命令、操作法などのほか、エラー処理、トラブル処理の対処法なども含む。利用者には無縁の内部仕様などの説明は省き、利用の局面で必要な知識や操作を中心にまとめたもの。マニュアルをハードディスク上に置き、オンラインマニュアルとして参照できるようにしてあるソフトも多い。

ユーザファイル[user file]

システムファイルに対し、主としてユーザが作成した応用ソフトウェアやそのデータを記録するファイル。データファイルともいう。その内容は

さまざまで、用途・目的・形式によってさらに分類できる。

ユースウェア[useware]

コンピュータシステムの利用に関する有料サービス、またはそのために必要な技術やノウハウ。コンサルティング、運用代行、ユーザ教育など幅広い支援サービスを含む。パソコンの爆発的な普及ににともなって、トラブルに対応できないユーザの支援が社会的な要請となった現状を踏まえて、1997年には生産者、ソフト開発者、販売者が協力する日本ユースウェア協会(JUA)が発足した。

優先順位①[priority]

現在の計算機システムでは、スループットの向上や資源の有効利用を図るため、マルチプログラミング方式を採用して同時に多数のジョブを処理している。これらの各処理にその必要の度合いに応じ資源の割当てを行う。その割当ての基準を優先順位という。優先順位には、ジョブ待ち優先順位や実行優先順位などがある。優先度ともいう。

優先順位②[precedence of operator]

複数の演算子間における評価の優先順位。(関)算術演算子

ユーティリティプログラム[utility program]

ユーザの処理を一般的に支援する汎用性の高い計算機プログラム。エディタや診断プログラム、追跡プログラム、分類プログラム、マージプログラム、ファイルコピーなどの媒体変換プログラム等の総称。通常は、コンピュータのメーカが作成して提供する。

(参)オペレーティングシステム、(参)サービスプログラム

ユニーク[unique]

一意的であること。同一名がなく、多数の中からあるものを特定できること。

ヨ

要求定義[requirements definition]

システムのユーザ、クライアントが、これから開発するシステムにどのような機能、性能、安全性、障害対策を要求しているかを把握、整理し、開発システムの機能、性能を設定すること。次のような点に留意して行う。

(1)ユーザ部門の率直な要求を知り得る協力関係を築く

(2)現状への不満を解消する具体的なニーズを把握する

(3)多くの要求を整理し、重点課題を明らかにする

(4)設定した開発システムが要求をどのように満たすかフィードバックする

(5) 新システム移行、運用にともなう業務の変更点に理解を求める

要求分析[request analysis]

現状分析により新システムの開発が必要と判断した場合、利用部門の要求案と期間・費用・人員などを配慮した開発計画案の双方をまとめること。要求仕様書や開発計画書を作り、利用部門との調整を図る。

呼出し命令[call instruction]

サブルーチンに実行の制御を渡す命令。コール命令ともいう。一般にサブルーチンからの戻り番地をスタックに格納して、サブルーチンの入口点に制御を移す。(対)リターン命令

予防保全[preventive maintenance：PM]

あらかじめ定めた期間内に定期的に点検や動作の確認を行い、障害を未然に防止するための保守作業。予防保全は障害の発生を抑えることとなり、MTBFを向上させ、システムの信頼性に寄与する。例えば、磁気ディスク装置などでときどき発生する読取り障害など、自動的にリカバリできたものについても、OSのログ情報を解析したうえで交換することが多い。(関)RAS、信頼性、保守

予約語[reserved word]

各プログラム言語の中で、特定の意味や用法をあらかじめ決めてあり、定められた書き方でしか使えない語や綴り。ユーザがシステム名、ファイル名、変数名などに使用することができない。

予約文字[reserved character]

システムやアプリケーションが制御のために専用する文字で、ユーザがファイル名やマクロ記述に使えない記号文字。

ヨ

ライセンス[license]

使用、占有、アクセスなどに対する許可、許諾。

（関）アクセス権、サイトライセンス

ライタ[writer]

ジョブスケジューラの機能のひとつ。出力待ち行列中から、優先順位の高いジョブを取り出し、そのジョブの指示により出力装置に出力データを書き出すプログラム。

ライフサイクル[life cycle]

ハードウェアやソフトウェアの導入から廃棄までの期間、またはその間の増設や変更の過程。業務処理などに、いったんコンピュータシステムを利用し始めると、それを使わなかった以前の状態に戻すことは不可能になる。より多くの仕事をコンピュータで処理する要求が生じ、より高性能なハードウェアの増設、より使いやすいソフトウェアへの改善がなされる。

またコンピュータ技術の進歩は速いため、次々と新しいハードウェアやソフトウェアが開発されてくる。このためユーザは3~5年程度でハードやソフトを刷新することが多い。廃棄されるシステムがあれば代替の新規システムが要求されるわけで、こうしたコンピュータシステムの誕生・成長・廃棄の流れをライフサイクルという。

ライブラリ[library]

本を集め、保管、利用に供する図書館（ライブラリ）にならい、プログラムの題目あるいはその要旨などを記入した資料やプログラム媒体を格納する保管庫などを示す。集めたプログラムそのものをライブラリともいう。プログラムを体系的に組織し、管理する人をライブラリアンという。

プログラムライブラリ（program library）は、補助記憶装置や媒体に記録したプログラムの集合。プログラマが必要に応じて取り出し、編集し、利用することができる。次の 3つに大別できる。

（1）ソースプログラム形式のまま記録したソースモジュールライブラリ

（2）再配置可能の形式で記録したオブジェクトプログラムライブラリ

（3）実行可能な機械語形式で記録したロードモジュールライブラリ

ラインプリンタ[lines printer：LP]

1行分の文字数と同じ数の印字機構をもった高速のインパクトプリンタ。一般にドットマトリックス方式より活字方式のほうが10倍ほど早い。行印字装置ともいう。印字は行頭から逐次に行うのでなく、1行分の文字データを受取り、その中でもっとも早く印字の機会を得た字から印字する。巨視的には、1行をいっせいに印字するとみなしてよい。普通、連

続用紙を用いる。印字速度は1分間に印字できる行数(LPM=lines per minute)で表す。

ラウンドロビン[round robin scheduling:RR]

TSSで、CPU時間(その使用権)を各タスクへ均等に配分する方法。各タスクの優先順位が同じ場合に有効。TSSユーザに公平なサービスができる。ジョブの投入順、タスクの生成順などの順序にしたがって各タスクに次々とCPU時間を分配する。ラウンドロビン方式では、一定時間(タイムスライス)が経過すると強制的にタスクの実行権を打ち切り、次のタスクに渡す。実行中のタスクが入出力命令を出すときにはCPU時間を放棄し、待ち行列の最後尾に回す。

(関)タイムシェアリングシステム

ラジオボタン[radio buttons]

WindowsやMacOSなどのGUIで用いる小さなボタン。これをマウスでクリックすることによって、機能や条件の切り替えを行う。主として複数項目の中からどれかひとつのモードのみを選択する場面で利用される。ラジオのAM/FM切り換えボタンなど、同時にはいずれかひとつのモードしか選択できないスイッチから比喩的に名付けられた。

ラベル[label]

①プログラム中で飛越し先などを示すためにつける標識名。モジュールや配列領域の先頭に名前をつけるときなどに用いる。goto文の制御の移る先の文、またはcase文、default文等にマーカとして付けられる標識のこと。ラベルの付いている文をラベル付き文(labeled statement)という。

②ボリュームやファイル等のデータの集合に識別用につけた文字列。ラベルはファイルの管理や処理の便宜のため使用し、その集合に関する情報を保持し、識別する。処理の対象に付ける名称で、ハード的な外部ラベルとソフト的な内部ラベルがある。

(1)外部ラベル:ファイル媒体(ボリューム)の外部に貼る名称記入用の用紙。ステッカーともいう。

(2)内部ラベル:電子的に書き込んだラベル。ボリュームに付けるものとファイルに付けるものがある。記憶媒体上のファイルの管理や処理、制御を円滑に行うために使用する。ISOでは、各媒体のファイル構造を制定しており、この中にラベルの書式も規定している。

(2-1)ボリュームラベル:磁気記憶媒体のボリュームの始めと終わりに付けるラベル。ボリューム内のファイルを識別する情報を記述する。ボリュームの先頭にボリューム通し番号とVTOCのアドレスなど記録ファイルの管理情報を記録する。

(2-2)ファイルラベル：ファイルの始めと終わりに付けるラベル。ファイルの識別情報・作成日情報などが書かれている。先頭にはヘッダラベルをつけて、ファイル制御の情報を記述する。ファイル末尾にはファイル終わりラベル(EOF)を記録する。ラベルもまたファイルを構成するレコードであるので、ラベルレコードとも呼ばれる。磁気ディスクでは、VTOCにすべてのファイルのファイルラベルが集中管理されている。(参)VTOC

ランダマイジング[randomizing] (同)ハッシング

ランダムアクセス[random access]

ファイルのアクセス法のひとつ。直接アクセスにほぼ同義。記憶装置からデータを得たり、記憶装置にデータを入れたりする機能で、その過程が前回アクセスしたデータとは無関係に目的のデータの記憶場所にだけ依存するもの。ファイルオープン後、ファイル上の位置を指定することによって任意の位置にアクセスすることができる。ファイルにランダムにアクセスできるかどうかは、ファイルの性質に依存する。COBOLでは、乱呼出しという。(類)直接アクセス、(関)直接編成ファイル、索引順次編成

ランダムファイル[random file] (参)直接編成ファイル

ランニングコスト[running cost]

システムの運用、設備や機械等の運転、稼働にともなって発生する費用。コンピュータシステムの場合、電気代や保守点検費、出力用紙、補助記憶媒体、インクリボンなどの消耗品代など。プログラムのメインテナンスにかかる費用を含むこともある。これに対し、導入時に一時的に要する費用をイニシャルコストという。

乱呼出し[direct access] (同)ランダムアクセス

リ

リアルタイム処理[real time processing]

コンピュータに処理要求の信号とデータを入力すると、ただちに処理し、出力や応答を行う処理方式。実時間処理、デマンド処理、即時処理ともいう。多くの場合、端末装置から通信回線を介してセンタのホストコンピュータにデータを入力し、処理結果の応答を受ける方式をとる。

リーダ[reader]

①ジョブスケジューラの機能のひとつ。JCLを入力し、ジョブ待ち行列に入れ、入力データを読み込み、イニシエータに通知する。

②入力装置の読取り装置。

リエンジニアリング[reengineering]

部品の再利用の際に、既存のシステムを調べ、新しい要求にかなうように変更すること。カスタマイズ(customize)が、既存システムの一部を修正することをさすのに対し、リエンジニアリングは、どこをどう直すべきかを知るための仕様分析の工程も含む。

リカーシブ[recursive]　(同)再帰

リカバリ[recovery]　(同)障害回復

リスク分析[risk analysis]

情報システムに潜む危険を把握する分析技術。日本情報処理開発協会(JIPDEC)はJRAM(JIPDEC Risk Analysis Method)を開発、普及に着手した。(類)リスクマネジメント

リスクマネジメント[risk management]

危険管理。情報処理の分野では、信頼性やセキュリティに関連したシステム全体の安全確保にともなう危険管理のこと。資産の価値とリスクの発生要因を分析して対策を講じ、最小のコストでリスクを最小に抑えることが目的である。(参)信頼性、セキュリティ

リストア[restore]

磁気ディスクやフロッピーディスクに保存(バックアップ)しておいたデータをハードディスクに戻したり、システムに読み込むこと。操作上のミスやプログラムのバグによって、システムにトラブルが発生したときなど、元の状態に復旧することができる。

リスト構造[list structure]

複数のデータがある順序にしたがって並んでいる構造。個々のデータの論理的な並びと主記憶装置内での並びが一致していない点で配列と異なる。順序を示す情報として各データはポインタを持つ。リスト構造のデータを操作・処理することをリスト処理という。

代表的なリスト構造には次のようなものがある。

(1)単リスト(線形リスト)(linear list／linked list)

データ要素の数が動的に変化する1次元の表で、アクセスがその両端に限られるもの。スタックは線形リストのうち一方の端からのみアクセスできるものである。

(2)双リスト(双方向リスト／連鎖リスト)(doubly-linked list)

(3)環状リスト(循環リスト／巡回リスト)(circular list)

単リスト、双リストは要素の数が動的に変化する 1次元の表である。単リストはアクセスをいっぽうの端に限るもので、スタックがこれに相当する。双リストは両端からのアクセスを許す。リスト構造を持ったデータ用

リ

のプログラム言語として、LISPやPROLOGなどがある。

リスト処理[list processing]

データ構造は配列とリストに大別できる。項目がある順序にしたがって並んでいる集合のことをリストという。リスト構造のデータを操作・処理することをリスト処理という。リスト構造を持ったデータ用のプログラム言語として、LISPやPROLOGなどがある。

リソース[resource]　(同)システム資源

リターン命令[return instruction]

サブルーチンや割込み処理を実行後、主ルーチンに復帰するために使う命令。サブルーチンや割込み処理でリターン命令を実行すると、スタックポインタの示すスタックの内容、すなわち直前の戻り番地を参照し、主ルーチンで分岐した直後の命令に戻る。

(参)サブルーチン、スタック、(対)呼出し命令

リファレンスデータベース[reference database]

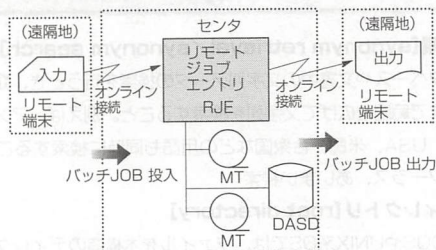
情報源の案内を主たる目的とするデータベース(文献データベース)。

データベースは当初、技術上の制約から情報がどこにあるのかを探す手段として始まり実用化に至った。リファレンスデータベースには文献のタイトルや著者名、掲載誌、内容の抄録などがある。検索した資料を直ちに入手できるファクトデータベースと連動するものが多い。

リミットチェック[limit check]

入力データの内容をチェックする方法のひとつ。データの長さや値の上限や下限があらかじめわかっている場合に、その条件を満たしているかどうかを調べること。制限チェックともいう。特にある範囲内にあるかどうかを調べることを範囲チェックという。

リモートジョブエントリ[remote job entry：RJE]

遠隔地の端末とセンタの端末を区別せず処理するセンタの計算機システムに、遠隔地の端末からジョブやデータを投入し、処理する方式。遠隔ジョブ入力ともいう。ジョブと入力データを一括してオンラインの端末装置から通信回線を介して入力し、ホストコンピュータがジョブ実行した結果をやはり通信回線を介して端末装置へ送り返す。データの送信や結果の入手に、郵送や輸送の必要がなくなるが、処理結果の受け取りにリアルタイム性は特に求めない方式といえる。

リモートバッチシステム[remote batch system]

遠隔地の端末とセンタコンピュータのバッチ処理システムを通信回線で結んだコンピュータシステム。遠隔地の端末が送ってくるジョブやデータを、センタのジョブやデータと区別せずに、センタの計算機システムで処理する。出力結果は端末で入手できる。プログラムやデータを郵送したり輸送したりする必要がなくなる。このような処理方式をリモートバッチ処理(remote batch processing)という。(関)オンラインシステム

リモートファイルアクセス[remote file access]

自システム内のファイルにアクセスするのと同様の操作で、ネットワークで接続した別のコンピュータのファイルを直接アクセス可能とする機能。実ファイルを持つ相手のコンピュータのことをファイルサーバと呼ぶことがある。UNIX、NetWare、Windows NTのネットワーク機能を持

つOSで、この機能を提供している。遠隔ファイルアクセスの具体例には、NFS、RFS、VFSなどがある。

リモートメインテナンス[remote maintenance]

ユーザシステムのハードウェアやソフトウェアの修理、点検およびソフトウェアの入れ替え、追加などを直接ユーザに出向かず、通信回線を接続した保守担当会社の端末から行う保守作業。遠隔保守ともいう。遠隔保守により、システムがまったく起動できない場合を除き、点検作業や障害の初期調査では保守員の移動時間が不要となり迅速な対応が可能となる。

リレーショナルデータベース[relational database : RDB]

(同)関係データベース、(関)関係代数

リレーション[relation] (同)関係

リンケージエディタ[linkage editor]

(同)連係編集プログラム

ル

類義語検索[synonym retrieval/synonym search]

データベースや文書中の文字列データの検索を行うとき、類似性をもった用語まで範囲を広げて文字列を検索すること。例えば、アメリカを指定すると、USA、米国、合衆国などの用語も同時に検索すること。
(関)シソーラス、あいまい検索

ルートディレクトリ[root directory]

MS-DOSやUNIX系OSでは、ファイルを木構造のディレクトリによって管理している。データの管理を木(ツリー)に例えると、ファイルは葉(リーフ)に、ディレクトリは根(ルート)や節(ノード)に相当する。木構造の最上位(根に相当)のディレクトリをルートディレクトリという。

ループ[loop]

①プログラム中で繰り返して実行する一連の命令。または、ある特定の条件を満たすまで繰り返し実行すること。繰り返し動作の打切り条件を正しく設定せず、プログラムが終了しない場合を永久ループ、無限ループ(permanent loop)、閉ループ(endlessloop)という。(類)繰返し
②一連の要素(機器やデータなど)を連ねた構造で、一周してもとにもどる道すじをループ(またはリング)という。

レイアウト[layout]

①ファイルに書込むレコードの属性やデータ長を考慮して、効率よく処理できるように設計すること。

②表示したり印刷する文書の文章や図表の配置。割付けともいう。ソフトの高機能化が進み、文字の書体、大きさや配色などが指定できるシステムでは、見栄えも含んでレイアウトという。

レグレッションテスト[regression test]

既存のシステムに修正、変更、機能追加をした場合、これらが正しく行われたことを確認するテスト。新機能部分だけでなく、従来機能のテストも行い、改変・追加したことにより、新たな誤り(レベルダウンという)が生じていないかを検証する。退行テスト、回帰テストともいう。

レコード[record]

相互に関連するデータ項目の集まり。データ処理は、レコードを単位として行う。論理レコードと物理レコードの二つに分ける。

①論理レコード(logical record):ファイルの構成要素で1件の事象に関連する項目(例えば個人情報なら、その個人の氏名・生年月日・性別・住所などのデータ)の集まり。プログラムでデータを取り扱うときの入出力の単位となる。
表計算ソフトにCSV形式のファイルを読み込ませる場合には、横1行分のセルが1レコードに対応する。ワープロソフトでは、表示の行数にかかわらず、改行コードまでの文字列が1レコードに対応する。

②物理レコード(physical record):入出力装置や補助記憶装置のファイルをアクセスするときに、実際の読み書きの物理的な単位として使用するデータの最小単位。磁気テープではIBGから次のIBGまでの1ブロック、磁気ディスクでは1セクタなどの単位。一般に、関連するデータの集合が1論理レコードを形成し、複数の論理レコードをブロック化して1物理レコードを構成し、複数の物理レコードの集合がファイルを形成する。
(関)ファイル、ブロック、

(対)論理レコード、(関)ブロック化係数

レコード型[record type]

データベース設計におけるE-Rモデルで、実体をある属性でグループ化した集合のこと。例えば、ある会社の社員であるという属性で作られた集合(社員ファイルに相当)は、実体(社員に相当)の属性データの並び(社員コード、氏名、生年月日、性別、所属など)といくつかの(社員数に相当)実現値の並び(9206123、合格太郎、19720315、男、営業部など)で表現できる。COBOLのレコードに相当する。

レコードタイプ[record type]

ファイルに記録するレコードの形式。レコードを形成する語または文字の長さ(record length)は、理論的にはいくらであってもよいが、実際にはどう扱うかによって処理の仕方が異なる。以下のように分類される。

(1)固定長レコード(fixed length record)

ファイル中の論理レコードが、すべて同一の長さに記録してある形式で、プログラム上も扱いやすい。常に一定の長さ以下のレコードを扱う場合には、そのハンドリングの容易さから固定長レコードを多用する。しかし、項目長さが異なれば空白文字を埋め込んで規定の長さにするため、記憶領域の利用効率は必ずしもよくない。

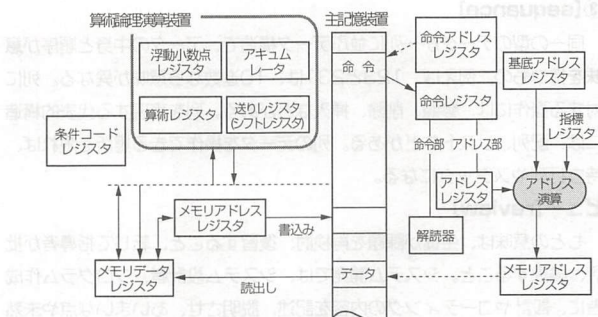
(2)可変長レコード(variable length record)

ファイル中の論理レコードの長さがそれぞれ異なるレコード形式。各レコードの長さが著しく異なる場合は、固定長レコードでは無駄が多くなるため、スペース効率の観点から可変長レコードをもつファイルを使用する。システムは、レコードの先頭にそのレコード長を、ブロックにまとめた場合にはブロックの先頭にブロック長を記録する。データの長さを記録する場所は、データの場所を示すカウント部の内部になる。索引順次編成の場合は、キー部を設け、アドレスに代えレコード内のキー項目で特定のレコードを探すことができる。

(3)不定長レコード(undefined length record)

可変長レコードと同様に論理レコード長がそれぞれ異なるが、おのおのの論理レコードが自分自身のレコード長情報をもたない形式。レコードの入出力はIBGをもとに行うため、ブロッキングはできない。

レジスタ[register]

中央処理装置の内部にあって、1語または数語、ときには数桁の情報を一時的に記憶する装置。特定の目的に使用し、随時、その内容を参照できる。処理中の命令、データ、アドレス、状態語などの一時記憶場所に使う。

ハードウェアとしては、フリップフロップ回路を複数個接続して、複数ビットのデータを一時的に記憶できる回路で構成する。

レジスタは使用目的によって、制御用レジスタ、演算用レジスタに分かれる。

(1) 制御用レジスタには、命令アドレスレジスタ(プログラムカウンタ)、メモリアドレスレジスタ、命令レジスタなどがある。

(2) 演算用レジスタには、アキュムレータ、インデックスレジスタ、基底アドレスレジスタなどがある。

(3) 汎用レジスタは使用目的を限定しないレジスタである。

レジストリ[registry／registry data]

①ハードやソフト、ネットワークなどの正当な使用者であることを登録すること、またはその登録簿に相当するデータベース。アクセス権の確認、課金情報の管理などに使用する。転じて、シェアウェアを使用する際に支払う寄付金など。

②Windows95/98では、ハードやソフトに関する各種設定情報を管理するデータベース。レジストリはユーザごとの設定情報を管理するＵＳＥＲ．ＤＡＴと、コンピュータごとの設定情報を管理するSYSTEM.DATの二つのファイルから構成されている。REGEDITというプログラムを起動してレジストリの内容を書き換えることができるが、むやみに変更するとオペレーティングシステムが動かなくなる恐れがある。

レスポンスタイム[response time]　(同)応答時間

列①[column／field]

表計算ソフトのワークシートで縦に並ぶ一連のセル領域。行の対語。

列②[row]

2次元配列や行列の縦方向の1列のこと。行(横に並ぶ1行)の対語。

(対)行

列③[sequence]

同一の型のデータが一列に並ぶデータ構造で、データの中身と順序が意味を持つもの。例えば、123と231は、10進数なら意味が異なる。列に対する操作には、参照、削除、挿入などがある。列を表現する代表的構造には、配列、リストなどがある。列のデータを操作できる場所を限れば、待ち行列やスタックになる。

レビュー[review]

もとの意味は、生徒が課題を再検討、復習すること、転じて指導者が批評、論評すること。システム開発では、システム設計者、プログラム作成者に、設計やコーディングの内容を記述、説明させ、あいまいな点や未熟な点を探り、自覚させること。レビューの場は、一般には複数のメンバが参加し、作成者の報告や発表の後、問題点の指摘、批判を行い、再検討や改良を促し、後工程に誤りを伝えないようにする場でもある。このほか、メンバ間の情報伝達、教育、進捗管理のねらいもある。(参)インスペクション

レプリケーション[replication]

データベースを他のサーバ上で複製する機能。分散型データベースシステムでは、あるデータを特定のサーバにのみ置くと、ネットワーク上のある部分にトラフィックが集中したり、ネットワーク全体のトラフィック量が増加したりする。そこで、データベースの複製をいくつかのサイトで持ち、一番近くのサイトのデータベースにアクセスするようにすれば、ネットワーク上の負荷分散と負荷の低減が可能となる。なお、更新の際一時的に複製間の内容の不一致が発生するため、レプリケーションには不一致状態の管理と定期的に内容を一致させる機構が必要となる。

連係編集プログラム[linkage editor：LKED／linker]

ユーザが書いたプログラムを実行可能なロードモジュールにするため、各種の計算機サブルーチンとユーザが作成したプログラムとを結合し編集するプログラム。結合編集プログラム、リンケージエディタともいう。一般に、オブジェクトモジュールは、アドレスの指定が0番地からとなっており、実行可能な形式ではない。また、他の副プログラムとのデータの授受の番地なども解決していない。実行可能にするためには、さらにさまざまな制御情報、呼び出される副プログラムや関数などの相互参照を解決し、可能ならば構成要素を再配置する必要がある。別々に翻訳したオブジェクトモジュールやロードモジュールから新たなロードモジュールを作成することが連係編集プログラムの役割である。

連鎖リスト[chained list]　(参)リスト構造

連続[consecutive]　(参)構造化定理、基本制御構造

連続ジョブ処理[continuous job control]

オペレーティングシステムのスケジューリング機能のひとつ。ユーザがコンピュータシステムに複数のジョブを与えると、OSがコンピュータ資源が遊ばないように優先順位を考慮したスケジューリングを行い、数多くのジョブの連続運転が可能になる。

連続用紙[continuous paper]

プリンタにセットする長い用紙。ストックフォームともいう。1ページずつ切り離したり、折りたたんだりすることができるように用紙にミシン目が入れてあり、左右には紙送りのための穴があいている。高速、大量の印字に適している。あらかじめ必要な項目を印刷した専用紙と、横線の入った用紙や白紙の汎用用紙がある。

ロ

ローダ[loader]

①周辺装置や補助記憶装置からプログラムをコンピュータの主記憶領域に読み込ませるルーチン。常時メモリ内に存在している。

②指定したオブジェクトモジュールおよびロードモジュール(ユーザ用プログラム)を作り上げ、そのプログラムを実行させるサービスプログラム。特に、連係ローダ(linking loader)と呼ぶこともある。

ロード[load]

一般に、データなどを処理するために格納場所から取り出して記憶すること。

①主記憶領域のデータをアキュムレータやレジスタに転送すること。

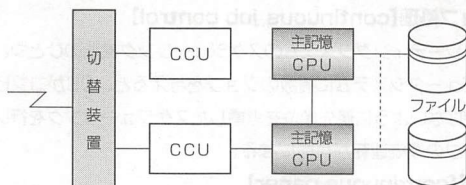
②プログラムやデータ等を補助記憶装置から取り出して、内部記憶装置に格納すること。OSがアプリケーションソフトを読み込むことはプログラムロード、アプリケーションがデータを読み込むことはファイルロードという。

③入力装置にデータ媒体をセットすること。例えば、外部記憶装置に記憶媒体を装着すること。(対)アンロード、セーブ

④負荷、または処理要求。

ロードシェアシステム[load share system]

負荷分散システム。ある業務に関して複数台のコンピュータを使い、その処理形態(オンライン、バッチ)や処理発生地域などで区別し、負荷を分担させるシステム。銀行のオンラインサービスは地域による分担をしてい

る。ひとつのシステムがダウンしても、他システムでカバーできる。

ロードファクタ[load factor]

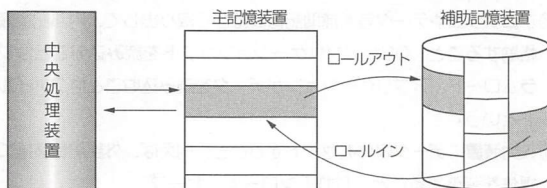
直接編成ファイルでは、レコードのキー値とレコードアドレスは1対1で対応すれば理想であるが、一般的には、重複(シノニム)ができたり、逆に空き領域ができたりして記憶装置の使用効率が低下する。エクステントの大きさとレコード数の間には適当な比率がある。そこで予め最大レコード数を想定して、多対1の対応でアドレスを求めたうえ、さらに余裕を持たせるとシノニムの発生が抑制できる。このエクステントの最大記憶容量に対する記録レコードの容量の比率をロードファクタという。約80%程度が適正であると言われている。

ロードモジュール[load module]

実行のために主記憶装置内へロードし得る計算機プログラムの単位。連係編集プログラムの出力がロードモジュールである。

(関)連係編集プログラム

ロールイン・ロールアウト[roll-in・roll-out]

階層構造をもつ記憶装置で、実行可能なプログラムやデータを補助記憶装置上に格納しておき、必要に応じて上位の記憶装置に呼び出し(ロールイン)、不要になったら、再び下位の記憶装置上に戻す(ロールアウト)という機能をもっている。特にページング方式の仮想記憶システムではページイン・ページアウトという。(類)ページング、スワップ、オーバレイ

ロールバック[roll back]

システム障害の発生で中断された処理を、その処理のすべてが行われなかった状態に戻すこと。マスタファイルのロールバックの場合、更新前情報を用いてロールバック処理を行う。この後、ロールフォワード処理を行う。

ロールフォワード[roll forward]

ファイル装置の障害などで破壊されたファイルを、障害発生の直前の状態にまで復元させること。セーブしたファイル（バックアップファイル）を正常なファイル装置に戻し、セーブ時以降の更新後情報（ジャーナルログ）でファイルをすべて更新し、復元を行う。

ログ[log] （同）ジャーナルログ

ログアウト[logout]

コンピュータシステムやネットワークのサービスの利用を終了させること。ログインからログアウトまでの状態をセションという。

ログイン[login]

コンピュータシステムやネットワークのサービスの利用を開始すること。アクセスを可能にするために、端末からIDとパスワードを入力して、システム資源と接続させる。端末からのアクセスの場合、通信ソフトの起動、プロトコルの設定、ダイヤリング、IDコード、パスワードの入力などの手続きをいう。

ロック[lock]

①ファイルアクセスを一時にはひとりに制限すること。

②ファイルにリードオンリー属性を付与すること。

論理エラー[logic error]

プログラム命令の組立て方に誤りがあるために生じるエラー。シンタックスエラーに対比して用いる語。論理的な誤り（設計段階の誤解、プログラミング段階の誤記、実行段階の誤操作など）が原因となる。論理エラーの検出には各種のテストを行う必要がある。

論理演算[logical operation]

記号論理学の規則に従う演算。数値を扱う数値演算と異なり、論理値は2値しかない。

論理積(AND)			論理和(OR)			否定(NOT)	
条件式1	条件式2	結果	条件式1	条件式2	結果	条件式	結果
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		

論理演算の基本公式($\overline{A}$はAの否定、・は論理積、＋は論理和)

- 交換律　A＋B＝B＋A　　A・B＝B・A
- 結合律　A＋(B＋C)＝(A＋B)＋C　A・(B・C)＝(A・B)・C
- 分配律　A＋(B・C)＝(A＋B)・(A＋C)　　A・(B＋C)＝A・B＋A・C

ロ

- べき等律 $A+A=A$　　$A \cdot A=A$
- 吸収律　$A+A \cdot B=A$　$A \cdot (A+B)=A$
- ド-モルガンの定理　$\overline{A+B}=\overline{A} \cdot \overline{B}$　$\overline{A \cdot B}=\overline{A}+\overline{B}$

論理演算子[logical operator]

数学やプログラム中で論理式を表すために用いる記号。論理変数をオペランドとし、複数の条件式をつないだり、真理値を逆転したりするときに用いる。

プログラム言語によって異なることもあるが、否定、論理積、論理和、排他的論理和等を示す。論理演算子の優先順位は、(1)否定、(2)論理積、(3)論理和の順であり、この順を変えたいときや明確に表現したいときにはかっこを用いる。かっこ内の演算の優先順位が高くなる。論理作用素とも呼ぶ。

論理		論理演算子の例				算用記号
否定	NOT	.NOT.	¬	¬	!	¬ ‾
論理積	AND	.AND.	∨	|	&&	∩∧+・
論理和	OR	.OR.	∧	&	||	∩∧+

論理演算命令[logical instruction]

記号論理学の規則に従う演算命令。アセンブラのAND、OR、NOT命令、論理式の真理値を求める演算など。プログラム内で論理演算は、条件の判断や組合せの表現でも利用する。

論理回路[logic circuit]

論理素子を組み合わせた電気回路、または電子回路で論理値(2値)の入力に対して論理値(2値)を出力する回路。論理回路は、次のふたつに大別できる。

(1)**組合せ論理回路**：入力値によってのみ出力値が決まる。NOT回路、OR回路、AND回路など。

(2)**順序回路**(記憶回路)：回路自体に状態を保持する。ラッチ回路やフリップフロップ回路など。

論理空間[logical storage]　(参)仮想記憶

論理シフト[logical shift]　(参)シフト

論理積[AND]　(同)AND

論理積演算[AND operation]　(参)論理演算

論理値[logical value]

非数値データのひとつ。論理変数、条件式のとり得る値で、値は1か0、真か偽、HighかLowなどの2値のいずれか。処理系によっては、ゼロと非ゼロとするものもある。

論理変数[logical variable]

2値しか有しない論理演算のための変数。値は1か0、真か偽、HighかLowなど排他的である。ビットに特定の意味をつけてコード化した数(例えば 0なら男性、1なら女性とか、0なら未婚、1なら既婚)は、データコードである。

論理レコード[logical operator]　(関)レコード

論理和[OR]　(同)OR

論理和演算[OR operation]　(参)論理演算

ワークエリア[work area]

プログラム実行中にシステムが中間結果の格納するために一時的に使用する主記憶(またはカレントディレクトリ)に確保し、実行後不要になった時点で削除する記憶領域。作業領域ともいう。

ワークシート[worksheet]

表計算ソフトで、表題、データ、数式などを入力できるセル(ます目)を2次元に並べた表形式の画面。画面は縦横に分割した表形式で、格子状の項目欄(セル)からなる。ひとつのセルは行と列の二つの名前で指定でき、一連のセルは、始点のセルと終点のセルで指定できる。セルに文字や数値などのデータを入力するほか、式や関数を入力して縦横の集計処理などをインタラクティブに行うことができる。スプレッドシート(spreadsheet)ともいう。表の縦横のサイズは、論理的に定義できるが、画面に表示できるサイズには限界があり、これをワークシートウィンドウという。

ワークファイル[work file]

プログラムの翻訳、連係編集、データの分類等を行うとき、それぞれの中間的なプログラムや中間的なデータを一時的に外部記憶装置に保存しておくための作業用ファイル。オペレーティングシステムが適当な名前をつけてファイル領域を確保し、使用後は自動的に削除する場合が多い。

ワークフロー[workflow]

業務の発生から終了までの仕事の流れ、または仕事の流れを決める規定や機構。ワークフローの改善で業務の効率化を行える。オフィスワークの中心が他部門との情報の交換、文書処理であることに注目し、情報の交換と処理にコンピュータネットワークを活用し、企業全体の生産性を高めようという構想に基づく。

ワークフローシステム[workflow system]

ワークフローをコンピュータシステムで実現すること、またはその技術、システム。業務上の連絡・会議などのコミュニケーションやスケジューリング、申請や承認などの認証、物品や金銭、伝票の授受を電子的な技術で実行するシステムをさす。例えば、ネットワークを利用した連絡や会議のためのグループウェア、CSCW(コンピュータ支援共同作業)システム、暗号利用の書類転送、認証システムなどをいう。

ワード[word]

一般的には、ある目的をもつひとかたまりの文字列。

①ハードウェアの分野では、記憶装置、演算装置、制御装置などでひとま

とめにして取り扱う情報の基本単位。計算機により長さ(ビット数)は異なる。制御用のマイクロプロセッサなどの 4ビット単位のものから、64ビット単位までさまざまである。通常、1命令または 1数値が 1語、または 2語に相当する。

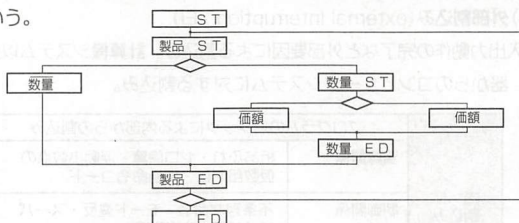
②オペレーティングシステムが、処理の状態を管理するために常に保持している情報。

③言語処理の分野では、プログラムやシステムで特定の意味をもつ文字列、約束。例えば、予約語、字句、パスワードなど。

④自然言語においては、単語、発言、キーワードなど。

⑤マイクロソフト社のワープロソフト、Microsoft-Wordの略称。

ワーニエ法[Warnier design method]

1970年代前半にJean-Dominique Warnier が提案したデータ構造設計法。データを集合としてとらえ、それが「どこで、いつ、何回」使われるかを明確化し、入力データの構造からプログラム構造を明らかにする。プログラム構造図は、入力データ構造の順次、選択、繰返しの表現としてそのまま反映される。LCP(logical construction program)ともいう。

ワーニングメッセージ[warning message]

コンピュータシステムからオペレータ、プログラマに出す応答メッセージ。プログラムの実行中、コンパイル中などに誤りの可能性がある状態を検出したとき、警告の意味で発するもの。

(参)エラーメッセージ

ワーム[worm]

コンピュータウィルスの一種。ネットワーク上で増殖して資源を占有し、トラブルを引き起こす。(類)コンピュータウィルス

ワイルドカード[wild card]

文字列やファイル名などのあいまい検索のときに使用する、任意の文字や文字列の代わりを務める特別な文字。任意の 1文字を表す「?」、任意の長さの文字列を表す「＊」などがある。トランプを使うゲームで、ジョーカやスペードのエースを他のカードと同等とみなすルールがあり、このカードをワイルドカードと呼ぶことに由来している。ワイルドカードの適用規

則は、OSやアプリケーションプログラムにより異なる。

ワクチンプログラム[vaccine program]

(類)アンチウィルスプログラム、(関)コンピュータウィルス

割込み[interrupt/interruption]

CPUがあるプログラムを実行中、それを一時中断して、ほかのプログラムを実行すること。プログラムの実行状況は、プログラム状態語(PSW)として、レジスタに格納されている。割込みが発生すると、PSWの内容を主記憶に退避し、PSWの内容を割込み処理用のプログラムの実行にかかわるものに置き換える。割込み処理が終了すると退避しておいたPSWの内容を再びレジスタに戻し、プログラムの実行を再開する。

割込みの要因は、大きく2つに分けられ、さらに表のように分類できる。

(1)**内部割込み**(internal interruption)

演算装置、記憶装置などの中央処理系で発生する割込み。実行中のアプリケーションプログラムがOSの機能に関わる要求やエラーを発生するなどの内部的要因によるもの。

(2)**外部割込み**(external interruption:EI)

入出力動作の完了など外部要因による割込み。計算機システム以外の機器からのコンピュータシステムに対する割込み。

内部割込み	プログラムのロジックによる内部からの割込み	
	演算関係	桁あふれ・ゼロ除算・浮動小数点の仮数部ゼロ・不当命令コード
	制御関係	不条理な命令・モード違反・スーパバイザコール(SVC)
	記憶制御関係	記憶保護違反・不合理な番地指定・ページ不在・セグメント未定義
外部割込み	計算機システム外からの割込み	
	誤動作関係	装置内エラー・電源異常
	計時関係	所定時間経過
	入出力関係	動作終了時(正常または異常)、状態の変化(用紙切れなど)
	他機器関係	他のCPUからの連絡、オペレータからの要求

一般に幾つかの割込みを随時受けつけることが可能であり、割込み処理中にさらに割込みがある場合は、優先度の高さによって処理する。また重要なプログラムの実行中は、特定の割込みだけを許すようにし、他の割込みを禁止することもできる。

A

Access (同)Microsoft Access

Ada

1980年にアメリカの国防総省が開発した、組込み型計算機システムのための高水準プログラム言語。1988年、ISOで規格化し、1991年、JISも規格化した。PASCALの流れをくみ、型(type)の宣言に特徴がある。信頼性が高く、モジュール化したプログラム開発が容易で、リアルタイム制御システムなど大規模なソフトウェアの開発に適している。また、言語の設計と並行して、Adaによるプログラム開発の支援環境(Ada Programming Support Environment：APSE)も規定している。Adaの名前は、イギリスの詩人バイロンの娘で、世界初の女性プログラマ、Ada Augusta Byronに由来する。

ALGOL[algorithmic language]

ヨーロッパの数学者が世界で共通に使えるアルゴリズム記述用言語作成の提案を行ったのを発端とし、各国の参加を得て標準化し、1960年ALGOL60文法として公表した。この言語体系をALGOL言語と呼ぶ。ALGOLはコンパイラ言語の一種で、PascalやFortranに大きな影響を与えたが、広く普及しなかった。

AND

論理演算子のひとつで、演算子の前後の二つのオペランド(条件式、論理変数)がともに成り立つときにのみ真となるもの。演算子は、AND、&、&&、AND(条件式1、条件式2)など。演算結果は、両辺の真理値が真(非0)の場合にのみ真を返し、それ以外は偽を返す。オペランドは論理変数に限らずビットを含む。(対)OR、(関)論理演算子

ANK[alphabetic、numeric and kana]

①コンピュータが扱う1バイト系の文字集合。英字、数字、カナ文字のこと。Alphabetic(英字)、Numeric(数字)、Kana(カナ文字)の頭文字を取った略称。一般にはその他の記号文字も含む。入出力は半角文字扱いである。アンク文字ともいう。

②上記に限ったデータを扱うこと。ANKしか扱えない端末をANK端末と呼び、漢字端末(日本語端末)と区別する。分散処理で端末がANKの場合、送信できるデータに制限が発生する。そのため、一般的に相手の端末がANK端末か否かをホスト側で端末属性のデータベースなどを用いて、あらかじめ知っておく必要がある。

ANSI[American National Standards Institute]

米国規格協会。米国における標準化を実行する団体。1928年に、電気、機械、金属など5つの団体の技術標準化委員会として設立。当初は

ASAと呼んだが、1966年USASIと改称し、1969年ANSIと改称した。米国の工業分野の標準化の最高機関で、国際標準化機構(ISO)、国際電気通信標準会議(IEC)の主要構成メンバである。規格原案は傘下にある組織(T1委員会など)が作成し、これを認証する形をとっている。データ通信関係では、X.12で示すEDIやX3T9.5で示すFDDIの標準化に力を入れている。

APL[A Programming Language]

IBMのアイバーソン言語(1962年、K.E.Iversonが定義した言語)をもとに開発したプログラム言語。会話型言語のひとつ。数学的な記述が簡潔に表現でき、新しい演算子を定義できる点に特徴がある。

APT[automatically programmed tools]

1956年、CAMI(computer aided manufacturing-international)の前身、APT協会が規格化した特殊問題向け言語の一種。NC工作機械を制御するプログラムを記述できるもので、加工する部品の形状を指定すると、工作機械を制御するコマンドに変換する。

ASCII[American Standard Code for Information Interchange]

ANSIが制定した米国標準情報交換用コード。ISOの7ビットコードにパリティビットを付加した 8ビットでコード化した文字集合(制御文字と通常文字の128種)。これをもとにJIS 8単位符号を制定している。

AVL木[AVL tree/Adel'son-Vel'skii-Landis tree]

平衡木の一種で、ある頂点から見て、左部分と右部分の枝の長さ(木の高さ、または深さともいう)の差が±1以内の木をさす。つまり、ある分岐点から見て左右の一方の枝のみが著しく長いということがない。

2分探索木の探索の理想形は完全2分木であるが、平衡木であれば探索の効率は悪くない。そこで2分探索木のバランスが悪くなり、探索の効率も悪くなったとき、木の作り直し(組み替え)をする。このとき完全2分木でないAVL木のほうが計算量が少なくすむ。ただし、木の高さが大きくなることは避けられない。(関)平衡木

B

B木[B-tree]

木構造の一種で、根からは1本以上、2m本以下の、各頂点からは最小m本(m≧3)、最大2m本の枝が出て、葉までの深さがすべて等しい木。これをm次のB木という。2分木に比較すると、ひとつの頂点から出る枝が多いので、データ量が多くなっても階層がさほど深くならない、平衡(バ

ランス)をとりやすいという特色がある。

BASIC[Beginner's All purpose Symbolic Instruction Code]

大型コンピュータのTSS用の会話型言語として1964年に米国のダートマス大学のJ.G.KemenyとT.E.Kurtzらが開発した汎用プログラム言語。Fortranに似た構造で、行列演算命令をもつのが特徴。1980年代には簡易版のBASICがパソコンの主流言語だった。本来はコンパイラ言語であるが、多くのパソコンではインタプリタ形式で使用した。1982年にはJIS規格として基本BASICを制定したが、1993年、新しいFull BASIC規格に改訂した。

BCC[block check character]　(参)チェックコード

BCS[block check sequence]　(参)チェックコード

BIOS[basic input／output system]

コンピュータの基本的な入出力装置を制御するインタフェース部分。ディスプレイ装置、キーボード、フロッピーディスク、プリンタなどの入出力制御ルーチンの集まり。ハードウェアの一部として組み込んであるROM-BIOSと、OSの一部として読み込むソフトウェア上に存在するものがある。BIOSは、機種によるハードウェアの違いをソフトウェアで補うことで、利用者の負担を軽減し、アプリケーションの生産性向上に資するなど、重要な役割を果たす。

BPI／bpi[bits per inch]

磁気テープにおける記録密度の単位。1インチ当たりのビット数で表す。800BPI、1600BPI、6250BPIなどのものがある。1列が9単位のテープでは、bytes per inchと解釈することもある。この場合、RPI(row per inch)なども使う。

bps[bits per second]

データ通信速度の単位で、回線または装置を経て1秒間に伝送できるビット数を表す(ビット／秒)。ボーという単位は送信するデータビットに対しての1秒間の変調回数を表す変調速度の単位で、通信速度の単位ではない。

英字

C

C[programming language C]

システムやアプリケーション開発に用いる代表的なプログラム言語のひとつ。簡潔な表現で効率のよいプログラムが書け、関数定義による言語の拡張が可能という特徴をもち、異なるシステム間でも互換性が高い。

1973年、アメリカのベル研究所のD.RitchieらがPDP-11のOS、UNIXを開発するために設計したシステム記述用のプログラム言語。1975年に発表して以来、大型計算機からパソコンまで多くの処理系があらわれ、ハード依存性が弱いことから広く普及した。データ演算子の数が多く、簡潔な表現で効率のよいプログラムが書けること、ユーザ定義による言語の拡張が可能という特徴をもつ。

C++[C plus-plus]

アメリカのベル研究所(現在のAT&T)のB.Stroustrupが、1983年に開発したオブジェクト指向型のプログラム言語。Cの仕様を含む拡張版。コンパイラやプリプロセッサのメーカにより仕様が多少異なる。

CA[control area]　(同)制御エリア

CAD[computer aided design]

コンピュータ支援設計。例えばLSIの設計、機械・建築・土木などの設計関係の作業をグラフィックディスプレイ装置、ディジタイザなどを用い、コンピュータと対話しながら行う。初期には作図、製図のシステムとして発展したが、その後、数値制御など機械と結合し(CAD/CAM)、さらに部品データベースや製品のモデリング技術とも結合し、CAE、CIMへと発展した。パソコンでも利用可能なソフトウェアが開発され、広い分野で利用している。(関)CAM

CAE[computer aided engineering/computer assisted engineering]

コンピュータ支援エンジニアリング。工業製品の設計段階で、技術計算、CAD、CAM、データベース、モデリングなどによりコンピュータを活用して計画、設計、解析、試作などを行うこと。
(参)ファクトリオートメーション

CAI[computer assisted instruction/computer aided instruction]

コンピュータ支援教育。コンピュータを利用した教授・学習システム。児童、生徒がディスプレイ画面の表示にしたがって課題を解決し、その結果の評価等を受けるという形の、個別学習指導を行う対話型のシステムの研究からスタートした。コンピュータが学校へ普及し、画像、映像の操作

も容易になり、児童、生徒の創造活動、情報収集、整理の道具として使用できるようになり、教科を問わず多方面に発展している。(関)CMI

CALS[commerce at light speed]

各種工業製品の開発、設計、受発注、流通等に必要な文書や図面などを、すべてコンピュータネットワーク上の電子データとして共有、交換するシステム。生産・調達・運用支援統合システムともいう。広い意味での電子商取引の基盤となることを目指している。

1985年、米国国防総省の軍事物資受発注システムCALS(Computer-Aided Logistics Support systems)から始まった。その後、民生分野でも導入され、利用範囲と機能の拡張によりその概念も変遷してきた。1987年にはCALS(Computer-aided Acquisition and Logistics Support)、1993年にはCALS(Continuous Acquisition and Life-cycle Support)に改称、1994年の改称で今日の名称に改められた。

CAM[computer aided manufacture]

コンピュータ支援生産。例えば、機械加工等をコンピュータによって自動化して、無人化を可能にすることなど。代表的な例には、NC(numerical control)工作機械や産業用ロボットを制御して加工すること。数値制御系は受け取った数値を解釈して動作するシステムで、プラント制御や工作機械に応用する。

CASE[computer aided software engineering]

コンピュータ支援ソフトウェア工学。ソフトウェアの生産性を高めるため、要求分析、定義から設計、製造、保守の過程をコンピュータを利用して自動化する技術、工学。ソフトウェア工学が提唱する構造化の概念、開発方法論、各種の支援ツール(CASEツール)の統合などは、システムの大規模化、高度化、複雑化、開発期間の短縮、省力化の要求に応える。

CASEツール[CASE tool]

CASEの目的を実現するソフトウェア。システム開発の作業を支援するソフトには、従来からプログラム設計ツール、プログラミングツール、デバッグツール、テストツールなどがあった。要求分析、定義、基本設計の段階(ウォータフォールモデルでいう上位工程)で用いる各種のツールもある。従来のツールと区別するときには、上流CASEツールともいう。対比的に従来のツールを下流CASEツールという。また広義には、運用・保守用のツール、ドキュメント管理、スケジュール管理などの作業を支援する共通CASEツールをも含めて、CASEツールと呼ぶ。

CASL

通商産業省が行っている情報処理技術者試験用のアセンブラ言語。COMETという仮想のコンピュータ上で動作する。CASLは4種類の疑似

命令、3種類のマクロ命令、23種類の機械語命令からなる。
(関)情報処理技術者試験、アセンブラ言語

CI[control interval]　(同)制御インタバル

CIDF[control interval define field]

制御インタバル定義フィールド　(参)制御インタバル

CJK統合漢字[China Japan Korean kanji set]

(関)UNICODE、UCS

CMI[computer managed instruction]

コンピュータ支援教授管理システム。学習の結果の記録、管理等を行い、学習指導の改善、教育研究等に資するデータ管理にコンピュータを利用すること。例えば、成績評価システムなどがある。

COBOL[Common Business Oriented Language]

事務処理用の共通言語として開発したプログラム言語。1959年9月に最初の文法ができた。最新のJIS規格は1988年に制定したものである。米国国防総省が中心になり、メーカやユーザが協力して開発した高水準プログラム言語であり、日常英語に近い形式でプログラムが書ける。改訂を重ね、長い間最もよく使われてきた事務処理向きの言語である。
(関)CODASYL

CODASYL[Conference on Data Systems Languages]

データシステムズ言語協議会。1959年、事務処理用共通言語の開発機関として発足。翌年COBOL60の仕様を発表した。計算機メーカ、ソフトメーカ、利用者の集りで、現在でもCOBOLの文法の保守・改定をしている。COBOLの仕様はCODASYLの承認のもとにANSI規格となり、これがISO規格となる。1967年、データベース言語の標準化を目指して、DBTG(Data Base Task Group)を組織、さまざまな提案を行った。1971年にレポートを提出して解散した後は、DDLC(Data Description Language Committee)が常設され、親言語方式や独立言語方式の併存できる仕様が検討された。CODASYL提案の仕様のデータベースシステムはCODASYL型データベースと呼ばれる。

CODASYL型データベース[CODASYL type database]

CODASYLのDBTGが提案したモデルに基づいたネットワーク型データベース。構成要素や用語は処理系によって異なるが、基本的には、(1)スキーマ定義言語、(2)サブスキーマ定義言語、(3)データ操作言語、(4)データベース管理システムなどで構成される。1987年ISO規格となり、同年JIS規格のNDLとなった。

CORBA[common object request broker architecture]

オブジェクト指向技術の標準化団体OMG(Object Management

Group)が、ソフトウェアの相互運用性(インターオペラビリティ)を図るために定めたオブジェクト管理プラットフォームの参照モデルの標準。CORBAに準拠すれば、アプリケーションを作成したプログラム言語が違っていても、またどのような分散アプリケーション環境でも、アプリケーション間でリクエストの発信と結果の受信を得ることができる。

CPS [characters per second]

字/秒、文字数/秒の略語。プリンタの性能、通信回線の容量などを示す単位。

CPU [central processing unit]　(同)中央処理装置

CSCW [computer supported cooperative work]

コンピュータ支援共同作業。LAN、ワークステーションなどの設備とグループウェアと呼ぶソフトウェアを利用して、主にオフィスにおける共同作業を行うこと。または、その環境。グループウェアの基本概念。具体的には、グループ参加者の個人スケジュールをコンピュータで管理した会議開催時間の最適スケジューリング、電子メールによる文書の回覧、閲覧、チェック、認証などを行う。グループウェアの適用形態として欧米で研究が進んでいる。(関)グループウェア

CSS [client server system]　(同)クライアントサーバシステム

CSV [comma separated value format]

アプリケーションソフトが扱うデータをテキストファイルとして入出力、記録する場合の形式で、フィールドとフィールドを半角のコンマ(,)で区切り、レコードとレコードを改行コードで区切る形式のこと。データを引用符(")で囲むか否かで三つに分かれる。

(1) CSV 1形式……フィールドのデータを引用符(")で囲む形式。
(例)"新・情報処理用語辞典","学研","933"

(2) CSV 2形式……フィールドのデータを引用符(")で囲まないでそのまま記録する形式。(例)新・情報処理用語辞典,学研,933

(3) CSV 3形式……フィールドの数値でないデータは引用符(")で囲み、数値データはそのまま記録する形式。K3フォーマットとも呼ぶ。
(例)"新・情報処理用語辞典","学研",933

CVFファイル [CVF file]

書き込み禁止、隠しファイル、システムファイルの属性をもつファイル。変更、削除するとシステムに異常をきたす重要なファイルをユーザの操作から遠ざけるためにCVFファイルとして特別に扱う。

英字

D

DAM[direct access method]

直接アクセス方式

DASD[direct access storage device]

(同)直接アクセス記憶装置

DB[database] (同)データベース

DBMS[database management system]

(同)データベース管理システム

DD[data directory] (関)登録簿

DD／D[data dictionary／directory]

データベースではシステムが保持するデータの意味を定めるメタデータの記録のこと。データディレクトリは、データを扱うために必要な制御情報を集めたもので、システムやプログラムが使用する。いっぽう、データディクショナリは、人間が定義した管理情報である。DBMSは、これらの制御情報と管理情報をデータとして扱うさらに上位のメタデータをもつことができる。これらの情報によりシステムは、多様な機能をもち、データの複雑な操作ができるようになる。

DDL[data description language] (同)データ記述言語

DDP[distributed data processing]

(同)分散データ処理システム

DES[Data Encryption Standard]

米国商務省標準局が1977年に制定した暗号化方式の規格。IBM社が開発した暗号化方式を、同局が標準として採用した。秘密性が必要な暗号化方式を規格化するのは異例のことである。この方式は、慣用暗号系の一種で、転置と換字の組合せによって暗号化し、暗号強度が高いのが特徴である。

メッセージを64ビットのブロックに分け、利用者が与える56ビットのかぎにより変換する操作を16回繰り返すことで暗号化する。復号するときはこの逆の手順を行う。DES暗号文をさらにDESで暗号化するというように3段に重ね、56ビットのかぎを112ビットに拡張するTriple-DESも使用されるようになっている。

DFD[data flow diagram] (同)データフローダイアグラム

DML[data manipulation language]

(同)データ操作言語

DPI／dpi[dots per inch]

印字装置や画像入力装置で入出力する情報のきめ細かさ(解像度)を示す単位。1インチ当たりの表示、または認識できる点の数で表す。

E

EBCDIC[extended binary coded decimal interchange code]

主に、IBM系汎用計算機で使用する文字コード体系。拡張2進化10進コードともいう。BCDコード(2進化10進コード)を拡張して2ビット追加し、8ビットで数字、英字、カナ、特殊文字を表現できるようにしたもの。256種類までの文字が表現できる。漢字を表現するためには2バイト必要とするが、1バイト系の文字と区別するために、シフト イン／シフト アウト コードを漢字コードの前後に挟んで使用する。(コード表は巻末資料)

EC[electronic commerce]　(同)電子商取引

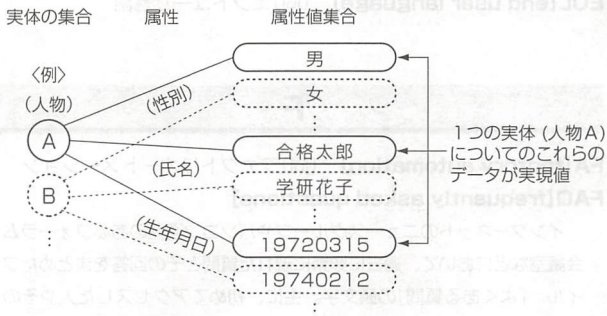
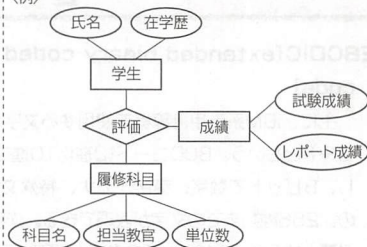
EDI[electronic data interchange]　(同)電子データ交換

EOR[exclusive OR]

二つのオペランドが異なる真理値をとるときに限り、結果が真になるブール演算。排他的論理和、非等価演算ともいう。オペランドは論理変数に限らず、ビットを含む。

ERR[error]　(同)論理エラー、シンタックスエラー

E-Rモデル[entity relationship model]

データモデルの一種。現実の世界を実体(entity)と関連(relationship)という二つの概念を用いて表現するモデル。データベース設計のためのモデルとして適しており、広く普及した。例えば、ある人物(実体)は、(性別、氏名、生年月日)などの属性によって、許される属性値の集合の中から具体的な(男、合格太郎、19720315)などの実現値と関連づけられる。

複数の異なる実態集合の間には、関連があって結び付けることができ、例に示すような図記号で表すことができる。これをE-R図という。

ESDS[entry sequenced data set]　(同)入力順データセット

EUC①[end user computing]

(同)エンドユーザコンピューティング

EUC②[extended UNIX code]

拡張UNIXコード。1985年に日本語UNIXシステム諮問委員会の提案に基づいてアメリカのAT&T社が定め、国際的に認められたマルチバイトの文字コード。非アルファベットを使用する国の文字コードにもUNIXを適応するため制定した。

日本語EUCの場合、漢字コードはJIS規格の漢字コードの最初のビットを1にした以外は完全に準拠している。ただし、半角カナにも2バイト分のデータが必要となる欠点がある。なお、現在国内のパソコンOSでデファクトスタンダードになっているシフトJISコードの漢字コード部分はJIS規格にまったく準拠しておらず、UNIXとパソコン間で日本語データを含む通信をする場合は、データ変換が必要となる場合が多い。

EUD[end user development]　(同)エンドユーザデベロップメント

EUL[end user language]　(同)エンドユーザ言語

英字

F

FA[factory automation]　(同)ファクトリオートメーション

FAQ[frequently asked questions]

インターネットのニュースグループやパソコン通信のあるフォーラムや会議室などにおいて、過去に頻繁に現れた質問とその回答をまとめたファイル。「よくある質問」の頭文字。主に、初めてアクセスした人やそのグループに新たに入ろうとする人が、グループ内の話題や関心事の概要を理解するため、またさまざまな疑問を解決するために利用する。参入者が増える可能性のあるフォーラムでは、SIGオペや発起人たちが質問を想定し

てあらかじめ作成しておく場合もある。

FAT[file allocation table]

記憶媒体上のファイルの管理情報を書き込む部分。ファイル自体の記録部分とは別の領域である。パソコンなどの外部記憶装置では、その格納領域の単位をクラスタという。ある大きなファイルが1クラスタでは収まらない場合は複数個のクラスタに記録するが、必ずしも連続した領域でなくてもよい。しかし、その情報を管理するFATが必要になり、ファイルの読込みの際は参照し、書込みをする場合は常に更新する。FATの構成は媒体によって異なるが、5インチ1メガバイトのフロッピーディスクでは、サーフェイス0、トラック35、セクタ24~26に存在する。

FAT32[File Allocation Table 32]

2ギガバイト以上のハードディスクをサポートするファイルシステム。従来よりも小さな単位(クラスタ)でドライブの領域を割り当てるため、ディスク領域をより効率的に使うことができる。Windows98の標準ユーティリティ「ドライブコンバータ(FAT32)」を使うことにより、従来のFAT16形式のファイルシステムをFAT32へ移行することができる。

FEP[front end processor] (同)フロントエンドプロセッサ

FIFO[first in first out]

先入れ先出し。サービスや処理を受けるために、待ち行列(queue)があるとき、最初に入ってきたものから処理する方法。(参)FILO、LIFO

FILO[first in last out]

先入れ後出し。後入れ先出し。サービスや処理を受けるために、スタックがあるとき、最も後から入ったものからサービスや処理をする方法。LIFO(last in first out)ともいう。(参)スタック、(同)LIFO、(対)FIFO

FIT[failure unit]

機器や部品の故障率を表す単位。1FITは、1時間以内に故障の発生する確率が1/10^9であることを示す。

FLOPS[floating operations per second]

1秒間に処理できる浮動小数点演算の数。コンピュータの処理速度の目安として用いる。通常MFLOPS(メガフロップス)を単位とする。スーパコンピュータは、ギガFLOPS級の能力がある。

FORTRAN/Fortran[Formula Translator]

科学技術計算用のプログラム言語。Fortranでは算術式に近い形でプログラムを表すことができる。最初に開発された高水準言語である(1956年)。言語の仕様はコンピュータの進歩と要求の高度化により何度も改訂して今日に至っている。最新のJIS規格は1994年に制定したFortran90である。COBOLと同様、標準化の進んだ言語である。スーパコンピュー

タ、並列コンピュータなどを利用した科学技術計算には、Fortranを多用しているが、新しいアーキテクチャに適したHPF(High Performance Fortran)と呼ぶ言語仕様が提案されている。

G

GIF[graphics interchange format]

米国のコンピュサーブ社が考案したフルカラー静止画符号化方式。WWWページのイメージデータ形式として標準的に使用したため有名になった。インターネット対応ソフトの多くにイメージのGIF変換機能やエディット機能が付加されている。256色までのカラーに対応できる。

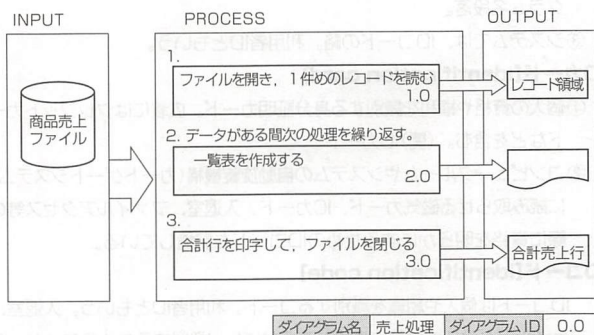
GUI[graphical user interface]

ユーザがコンピュータを操作するとき、画面に表示しているメニューやアイコンをマウスなどで選択することで処理を容易に行えるようにした操作環境。従来の多くのシステムでは、コンピュータの操作はオペレータがコマンドをキーボードから入力する方式、すなわちキャラクタユーザインタフェース(character user interface：CUI)が一般的であった。GUIでは、画面上に機能や内容をイメージできる絵などを表示し、コマンドを知らなくても操作が類推できるなど優れた点が多い。ハードウェア、ソフトウェアに大きな負担がかかるため、処理スピード・容量などの点に問題があったが、技術革新で高速なCPU、大容量のメモリが開発されたため、ユーザフレンドリなヒューマンインタフェースのひとつとして広く普及した。これにともなってGUI構築ツールが発達し、部品化の技術を取り込んでビジュアル開発環境が実現した。

H

HEX[hexadecimal]　(同)16進数、16進法

HIPO[Hierarchy plus Input Process Output]

ソフトウェアの機能を、入力(input)・処理(process)・出力(output)に分けて階層構造に表示する図形化方式の手法。ソフトウェア開発時に、文書化の手法として、また、設計の補助手段として用いる。プログラム内のモジュールを木構造で表した内容一覧表、各モジュールの機能概要を書いた総括ダイアグラム、および機能詳細を書いた詳細ダイアグラムから成る。CASEツールがとって代わりつつある。

HPF[High Performance Fortran]　(参)Fortran

HTML[hypertext markup language]

スイスのCERNが開発したハイパーテキスト記述言語。インターネットのWebページを記述できる。言語仕様は、文書標準記述言語SGMLのサブセットになっている。HTMLは、参照文書、静止画、動画、音声などの入っているファイルを指定(タグ付け)できるようになっており、対応するページを画面に出力するとそれらがハイパーテキストとして表示できる。ワープロや表計算ソフトなどの出力結果をHTMLに自動変換し、そのままWebページにするソフトもある。

Hz[Hertz]　(同)ヘルツ

I

英字

IBG[interblock gap]　(参)ブロック

ICカード[integrated circuit card]

ICメモリを内蔵したカード状の記憶媒体。磁気カード、フロッピーディスク(FD)に比べ記憶容量が大きく、書き換え可能などの利点も多く、磁気カードやFDに代わる新しい記憶媒体として実用化が進行中である。マイクロプロセッサを内蔵し、マイコン機能、インタフェース機能をもったカードもある。

ID[identifier／identification]

①Fortranの用語。名前。英字名、COBOLでは一意名。変数・配列・手続きなどを識別するために用いる文字列。(同)識別子

②COBOLの用語。Identification divisionの略。見出し部。プログラム名、プログラム作成者、作成日等を記入する。PROGRAM-ID.はプロ

グラム名段落。

③システムでは、IDコードの略。利用者IDともいう。

IDカード[identification card]

①個人の資格や権利を識別する身分証明カード。広義にはクレジットカードなどを含む。(関)IDカード

②コンピュータルームやシステムの自動旋錠機構(カードゲートシステム)に読み取らせる磁気カード、ICカード。入退室、ファイルアクセス等の際に資格を明らかにするためのIDコードを記憶している。

IDコード[identification code]

IDコードは個人や組織を識別するコード。利用者IDともいう。入退室、ファイルアクセス等の際に資格を明らかにし、登録するためのコード。IDカードや暗証番号に代わる本人確認の技術として、当人の指紋、声紋、虹彩(瞳の中の模様)など属人性が高く複写できない特性を利用する研究も進んでいる。

IEEE[the Institute of Electrical and Electronics Engineers Inc.]

米国電気電子技術者協会。1963年に設立。アメリカの電気、電子、通信、コンピュータなど幅広い分野の研究者・技術者の団体。世界最大の電気電子技術に関する学会で、全世界に10の地域本部があり、会員数は全世界で30万人を超える。

日本には、アジア太平洋地域本部に属する東京支部があり、東京支部に属する会員は 7千人を超える。IEEEは、技術分野別にコンピュータ、通信、放送技術、信号処理など36のソサエティを設けている。標準化活動として電子、電気部品、通信用バスコネクタ、LANなどを対象に幅広い活動を行っている。

IEC(国際電気標準会議)インタフェースバスともいう。

IRG[interrecord gap]　(関)ブロック化

ISAM[indexed sequential access method]

(関)索引順次編成

ISDN[Integrated Services Digital Network]

ITU-Tが提案した電話、データ通信、ファクシミリ、ビデオテックスなど性格が異なる情報サービスを1本の回線に統合する統合ディジタル通信網。サービス統合ディジタル網ともいう。通信サービスの国際標準とみなすことができる。目的はユーザ・網インタフェースを一元化し、高速で高品質な通信を安価な料金で提供し、将来への拡張性を保証することである。日本では同様なものとしてNTTが開発したINSがあり、国際化に適応できるようISDNとの整合性を検討している。

ISO [International Organization for Standardization]

国際標準化機構。産業界の諸技術の標準化ととりまとめのために1947年に設立した国際機構。電気一般を除く情報処理システム、工業製品を対象に、技術の発達、標準化を進めることを目的とする。本部はジュネーブ。160あまりの専門委員会を有し、標準化の課題を検討している。専門委の決議した標準案を全会員に示し、多くの賛成が得られれば国際規格となる。日本からは、JISC（日本工業標準調査会）が加盟している（1952年）。情報処理分野に関しては1987年からIECと合同組織（ISO／IEC JTC 1）となり標準の作成を進めている。最も有名な標準にOSI（開放型システム間相互接続）がある。

ISOコード [ISO code]

1968年に、ISOとCCITT（現：ITU-T）が勧告した国際的な標準コード。国際的なデータ伝送などの普及につれて、計算機の種類や各国の規格で異なる数字や文字などのコード体系を統一する目的で標準化した。日本のJISコードもこれに準拠している。紙テープやデータ伝送に用いる7単位コードと、磁気テープなどに用いる8単位コードがある。

ITU [International Telecommunication Union]

国際電気通信連合のこと。電気通信の分野で幅広い責任を有する政府間国際機関。1934年にそれまでの国際電信連合を発展させ、発足。本部をジュネーブにおく。ITUの目的は、電気通信技術の発達とその効率的な運用の促進、またそのために各国間の調整である。標準化組織として、従来のCCITTとCCIRの活動を継承した電気通信標準化部門（ITU-T）と無線通信部門（ITU-R）をもつ。

ITU-T [ITU-Telecommunication Standardization Sector]

ITUの常設機関のひとつで、初めはCCITTと称した。1993年7月、CCIR（国際無線通信諮問委員会）の一部を統合しITU-T（ITU-電気通信標準化部門）と改称した。無線を除く電気通信に関する技術、運用および料金についての研究を担当し、作成した標準を勧告として発表する。構成員はITUの構成員の中から決定する。15の専門委員会（SG委員会）から構成している。主な勧告には、ISDN関連のIシリーズ勧告、OSI関連のXシリーズ勧告、信号方式に関するQシリーズ勧告、モデムに関するVシリーズ勧告などがある。

J

Java

アメリカのサンマイクロシステムズ社が1996年に開発したオブジェク

ト指向言語。1991年に携帯情報端末のアプリケーション記述用言語としてC++をベースに開発したOak言語から発展した。Objective-CやSmalltalkなどの言語仕様も取込んでいるが、ほぼC++同様にプログラミングができる。Javaはネットワーク上でダウンロードするアプリケーションソフトを開発するのに適しており、インターネット上で共通のプログラムを実行するための標準的な言語になりつつある。ソースコードを中間コード(Javaアプレット)にコンパイルして実行するインタプリタ言語である。

1996年1月に正式版を公開。OSの違いを意識せずに開発可能、オブジェクト指向、動画の取扱いが容易になるなどが主な特徴である。

JCL[job control language]

コンピュータシステムが能率よく稼働するために、制御プログラムに実行すべきプログラムを指定し、必要なコンピュータ資源(記憶域、入出力装置とデータ)の指定、競合がある場合の優先順位などを指示する言語。ジョブ制御言語ともいう。(参)ジョブ、ジョブステップ

JIS[Japan Industrial Standards]

日本工業規格。日本の工業製品の標準化を目的に1949年制定の工業標準化法により、日本工業標準調査会(JISC)の審議を経て制定した国家的規格。情報処理技術分野では、文字コードやデータコードなど多くの規格を制定している。

JIS.7単位符号[JIS 7 bit character set]

JISで制定したコード(X 0201)のひとつで、7ビットで文字を表現する。これは、英数字、*、¥、%などの記号、伝送制御文字などの機能文字で構成する。ネットワークで接続したコンピュータ間のデータのやりとりに用いている。そのとき、1ビットのパリティビットを付加し、8ビットで通信することが多い。

JIS.8単位符号[JIS 8 bit character set]

JIS 7単位符号にもう1ビット追加し、8ビットで文字を表現したコード。7単位で表現できる文字に加えて、カタカナを使用できる。

JISコード[JIS code]

①JIS規格で定めた情報交換用の文字コード。ISO規格に準じて1976年に制定。7ビットで表現する7単位コードと8ビットで表現する8単位コードの2種がある。

日本語処理に必要な漢字コードはJIS情報交換用漢字符号の名で1979年に規格化した。

②JISで定めたデータコード。アプリケーションソフトウェアでしばしば用いる各種コードについて規格化した。都道府県コード、市区町村コー

ド、産業分類コード、勘定科目コードなどがある。

JIS情報交換用漢字符号[JIS kanji character set code]

JIS 8単位符号で表した文字に加え、漢字までも表せるようにした文字コード。漢字は16ビット(2バイト)を使い、漢字以外は8ビット(1バイト)を使う。そのため、漢字以外の文字から漢字へ入るときと漢字から漢字以外の文字に戻るときに、それぞれ漢字イン(KI)、漢字アウト(KO)の漢字シフト文字を挿入する。

漢字と非漢字(記号、英数字、ひらがな、カタカナ)などの2バイト文字(全角文字)を表現するために、コンピュータ用に割りふった16進数字4桁のコード(符号)。JIS規格では、漢字の使用頻度に応じて、JIS第1水準(漢字2965文字、ひらがなとその他の各種文字753文字、合計3418文字)、JIS第2水準(旧漢字など3390文字)、さらに補助漢字5801文字を指定した。通常のコンピュータ処理では、約7000字の文字があれば十分である。ほとんどのビジネスパソコンでは、第1水準、第2水準の漢字を標準装備するものが多い。(参)シフトJISコード

JPEG[joint photographic coding expert group]

①ISO/IEC JTCのSC29とITU-T共同でパッケージメディアの静止画像圧縮技術の標準化を進めている作業部会。

②同作業部会が定めたフルカラー静止画像のディジタル圧縮に関する標準。フルカラー静止画像符号化方式ともいう。1992年にISO/IEC標準、1994年に国際標準となった。

議論の過程で2値画像(白黒の文字や絵)に対しては不利とわかり、2値画像についてはさらに圧縮率の高いJBIG(Joint Bi-Level Image Coding Expert Group)が設立され、この方式も1993年にISO/IEC標準となった。

(参)MPEG

K

KSDS[keyed sequential data set/keyed sequenced data set]

(同)キー順データセット

L

LAN[local area network]

小規模な通信ネットワークの一形態で、同一敷地内、同一構内、同一建物内など比較的狭い地域に分散して配置したコンピュータやワードプロセッサ、ファクシミリ、電話などの多数の端末を伝送媒体で接続して構成した私設のネットワーク。企業内情報通信網、構内通信網ともいう。端末装置間の相互通信を可能にし、大量データの伝送、高速なデータ伝送、負荷の分散、資源の共有など高度なOA化の実現を目的とする。広域網(WAN)を一般の電話網に例えるならば、これは屋内の内線電話にあたる。ネットワークトポロジとしては、バス型、ループ型、スター型が代表的。元はUNIXを中心とするミニコン、ワークステーションのネットワークであった。

LANの適応可能分野の広がりで、技術の動向は、(1)より低価格で構築できる大衆化の方向(パソコンLAN)と(2)LANの長さや設置端末の増大による大規模化の方向とに分かれようとしている。

LIFO[last in first out]

後入れ先出し法。広義には待ち行列の手法のひとつで、先に発生した事象から順に格納し、とり出すときは逆に最も新しく格納したものから順に取り出すこと。情報処理の分野では、例えばサブルーチンの戻り番地の管理や多重の割込みを処理する入れ子の構造のデータを処理するために、後入れ先出しの記憶装置(スタック)が必要になる。(関)スタック

Lisp[List Processor]

1960年代にマサチューセッツ工科大学のJ.McCarthyらが開発した代表的なリスト処理言語。リストとは、文章や数式や定理等の非数値データをうまく表す並びのことで、順序をもっている。リスト処理とは、これらの要素の並べかえ、追加、削除等の操作である。LISPは人工知能や計算論理学の世界で広く使う。(関)リスト処理

LP[lines printer]　(同)ラインプリンタ

LPM[lines per minute]

行/分。ラインプリンタの印字速度の性能を表す単位。

LRU[least recently used]

最低使用頻度。最も長い間使わなかったという意味。仮想記憶システムで、主記憶装置に補助記憶装置からプログラム等を呼び出す際は、代わりのページを追い出す。この追い出すページを決定する方法のひとつで、最も長い時間使用しなかったページを出す方式。(関)仮想記憶

LSB[least significant bit]

2進数のデータの最下位のビット(桁)またはその内容。

M

MDR[memory data register]　(同)メモリデータレジスタ

Microsoft Access

マイクロソフト社が開発したWindows用のリレーショナル型データベースソフト。表計算ソフトExcelとの連携機能が大きな特徴となっており、Excelのワークシートを直接呼び出して、Accessのテーブルと同様に扱うことができる。追加変更されたデータは、そのままもとのExcelのワークシートにも反映される。一般にリレーショナル型のデータベースは、カード型のデータベースよりも習得がむずかしいとされているが、Accessにはサンプルやテンプレート、各種ツールを用意して簡易化するなど初心者に対する配慮がなされている。

MIDI[Musical Instrument Digital Interface]

シンセサイザや電子ピアノなどディジタル楽器を互いに制御するための標準的なインタフェースやその規格。1983年に日本が提唱した規格を、日米の電子楽器メーカが協議し合意に達したもの。IN、OUT、THRU(通過)の 3種類の端子をもつものが一般的。専用のMIDIケーブルで容易に接続できる。

もともとは電子楽器どうしを結ぶことを想定していた。現在ではパソコン用のインタフェースボードが多数あり、画面上からマウスなどで入力した楽譜を演奏させたり、ボードにMIDIケーブルで接続し楽器のキーボードから音楽データを取り込んだり多様な使い方をする。

MIME[multipurpose internet mail extension]

インターネットの電子メールで、英語以外の言語やマルチメディアデータを扱えるようにするために、インターネット活動委員会(IAB)が1992年に提唱した規格。お互いにMIME準拠のメールソフトを使用すれば、インターネットとつながる利用者どうしでマルチメディアデータを含む通信が可能となる。また、タイトル、名前、本文に各国の文字セットのコード体系を利用しても、文字化けすることなく利用できる。

MIPS[million instructions per second]

計算機の処理速度評価法のひとつ。中央処理装置が1秒間に実行できる代表的な命令(例えば加算命令が全体の95%、乗算命令が5%)が何百万回

かという値で、計算機の処理速度の目安とする。(参)FLOPS

MPEG[Moving Picture Experts Group]

①ISO/IEC JTC1のSC29で、パッケージメディアのフルカラー動画圧縮技術の標準化を進めている作業部会。

②同作業部会が定めた動画圧縮形式の標準。動画像符号化方式ともいう。最初の標準は1992年に国際標準となった。現在はMPEG1とMPEG2の二つの規格を制定している。なお、拡張も研究中である。

(1)MPEG1:CD-ROMドライブやN-ISDNに対応しており、150キロbpsから最大1.5メガbpsのデータ転送速度で、352×240ドットの画面サイズ、毎秒30コマの動画を再生することができる。画質的には家庭用VHS並みの品質である。

(2)MPEG2:B-ISDN、ハイビジョンテレビ、DVDにも対応し、アメリカやヨーロッパのディジタルTVでの採用が決まっている。データ転送速度や画素数によって何段階かのレベルに分かれている。(参)JPEG

MS[main storage] (同)主記憶装置

MSB[most significant bit] (対)LSB

MTBF[mean time between failures] (参)信頼性評価尺度

MTTF[mean time to failure] (参)信頼性評価尺度

MTTR[mean time to repair] (参)信頼性評価尺度

N

英字

NC制御[numerical control:NC]

数値情報によって制御すること。数値制御系は受け取った数値を解釈して、動作するシステムで、プラント制御や工作機械に応用する。

NDL[network database language]

ネットワーク構造のデータベースを定義・操作するデータベース言語のひとつ。NDLは、CODASYLのDBTG提案の言語仕様NDL(network database language)をもとにNDRが規定され、1987年にISO規格、JIS規格になった。次の3つのデータベース言語の構文と意味を規定している。

(1)スキーマ定義言語:データベースの論理構造を定義する命令文の集まり。

(2)サブスキーマ定義言語:適用業務プログラムのビューによるデータベース部分の論理的定義

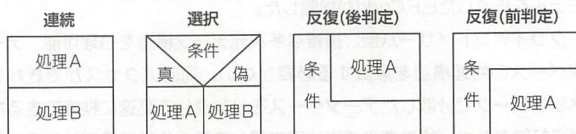
(3) **モジュール言語とデータベース操作言語**：適用業務プログラムがDBMSを利用するための定義

NOP [no operation]

次の命令に進むだけで、何もしない命令。単に命令のサイクルを遅らせるために用いる。

NOT

論理演算子のひとつ。条件式の前に記述し、その条件式が成り立つときには偽を、成り立たないときには真を返す。真理値を逆転させる演算子である。

NSチャート [Nassi-Shneiderman chart]

T.NassiとB.Shneidermanが作った構造化プログラミング向きの処理手順の図式表現。連続・選択・反復構造を、図のように表す。

null [null]　(同)ナル

O

O記法 (O notation)

アルゴリズムの計算量を示す表現法。*O*は、オーダと読む。

①***O*(オーダ)**：データサイズnが2倍、3倍、4倍、…、となると、実行時間も2倍、3倍、4倍、…、となるアルゴリズムの計算量は$O(n)$。

②**整列の計算量**：データ個数n

選択、バブル、挿入ソート：$O(n^2)$＞シェルソート：$O(n^{1.25})$＞マージ、クイックソート：$O(n\log n)$　※クイックソートの最悪は$O(n^2)$

③**探索の計算量**：データ個数n、[ ]は小数点以下切捨てを表す

線形探索：$O(n)$　平均比較回数：n／2回、最大：n回

2分探索：$O(n\log n)$　平均比較回数：$[\log_2 n]$、最大：$[\log_2 n]+1$

OCT [octal]　(同)8進数、8進法

ODBC [open database connectivity]

アメリカのマイクロソフト社が1991年に提唱したWindowsアプリケーションプログラムがさまざまなデータベースにアクセスするための共通APIの仕様。SQLによるアクセスを基本としている。現状では、データ

英字

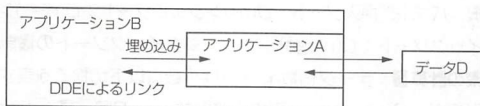
ベースソフトのメーカが互いに独自の仕様を公開し、他社のソフトで作ったデータベースでも検索できる各種のドライバを供給していたが、次第に統一したいという気運が熟してきた。ODBCのAPIを使用すれば、アプリケーションプログラムはExcelのデータファイル、SQL Server、dBASE、Oracle Serverなどのデータベースにアクセスが可能になる。製品には、マイクロソフト社のAccess、パワーソフト社のPowerBuilder、日本センチュラのSQLWindowsなどがある。

OLAP[online analytical processing]

エンドユーザが直接、基幹データベースを操作して問題点や解決策を見つけ出せるような分析型アプリケーションの概念。リレーショナルデータモデルを提唱したE.F.Coddが提唱した。

クライアント/サーバ型、複雑な多次元データ構造を処理可能、データベースの物理構造を意識する必要がない、同時アクセスができかつネットワークで分散したデータベースをリンクして高速に検索できるなどの特徴をもつ。経営者の意思決定支援システムや役員情報システムなどの構築に向く。なお、OLAPを構築できるシステム環境をデータウェアハウスと呼ぶ。

OLE[object linking and embedding]

Windowsアプリケーションプログラム間のデータのやりとりやデータの共有を実現させる機能。複数のアプリケーションソフトが同時に、それぞれのデータをやりとりし、操作性を向上させている。画面上のオブジェクトに他のオブジェクトを貼り付けたり、リンクで結んでアプリケーション機能の組み合わせが容易にできる。例えば、画面上のあるデータを示すオブジェクトをプリンタ出力のオブジェクト上にドラッグアンドドロップで持っていくと、それだけでそのデータを自動的に最適の形式でプリントできる。また例えば、文章ファイルの中に画像データファイルの情報を取込み、表示できる。

OLTP[online transaction processing]

(同)オンライントランザクション処理

OR

論理演算子のひとつで、演算子の前後の二つのオペランド(条件式、論理変数)の少なくともいずれか一方が成り立つときに、真になるもの。演算子は、OR、||、OR(条件式1、条件式2)など。演算結果は、両辺の真

理値が偽(0)の場合にのみ偽を返し、それ以外は真を返す。オペランドは論理変数に限らずビットを含む。

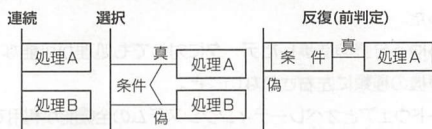
OS[operating system]　(同)オペレーティングシステム

OSI[open systems interconnection]

開放型システム間相互接続。ISO(国際標準化機構)が推奨するデータ通信のサービス定義とプロトコル仕様を総合的に規定したネットワークアーキテクチャの国際標準。標準は、OSI参照モデルを規定し、各階層ごと、各応用機能、ネットワーク管理機能などにわたる膨大なものとなっている。ITU-Tが積極的にOSIに関する勧告をXシリーズとして行い、順次ISOの国際標準となり、JIS規格になっている。通信回線による異機種間の接続は標準的なプロトコルで定めているが、データリンクの域を出ていない。これに対しOSIは、最終的にはアプリケーションまで接続可能となるような接続を目指す。

P

PAD[problem analysis diagrams]

1970年代後半に考案した処理手順の図式化手法のひとつ。基本制御構造は次のように表現する。問題分析図ともいう。

Pascal

チューリッヒ工科大学のN.Wirth教授が1975年に設計したALGOL系の言語。系統的なプログラミングの設計やアルゴリズム記述に適している。プログラミングの入門言語としてよく利用する。1990年にJISでも規格化した。

PC[program counter]　(同)プログラムカウンタ

PCMCIA[Personal Computer Memory Card International Association／Personal Computer Memory Card Interface Adapter]

パソコン・メモリカード国際協会の定めたパソコン標準規格、またはこの規格で標準化されたデバイス用のアダプタ。PCMCIAは、1989年日米の複数のメーカにより規格策定のために構成した団体で、ノートパソコン

英字

にメモリ、モデム、ハードディスクなどの周辺機器を容易に接続、交換できるように拡張バス、スロットなどについて規格を定めた。

PEM[privacy enhanced mail]

インターネット技術特別調査委員会(IETF)が標準化したセキュリティを強化した電子メールの標準。暗号技術を導入し、本文の盗聴を防止でき、電子署名により本文の改ざんも検出できる。また、公開かぎ証明書を利用した利用者の認証も可能。ただし、テキストデータしか送信できない。そのため、インターネットを経由したクレジットカードによる電子商取引(EC)やサイバーキャッシュ(電子現金、または電子マネー)などをやり取りする場合に利用できる可能性がある。最近では、PEMに準拠した電子メールソフトやサーバソフトが提供され始めている。

PGP[pretty good privacy]

米国のフィリップ・ジマーマン(Philip Zimmermann)が考案した暗号化電子メールプログラム。北米におけるインターネット上の暗号化電子メールの業界標準のひとつ。公開かぎ暗号化(RSA)方式を使用しているため、ディジタル署名にも使用できる。UNIX、VMS、MS-DOS、OS/2、Macintoshなどの環境で動作する製品がそれぞれ存在する。

PL/I [Programming Language one]

事務計算、科学技術計算等を問わず、すべての目的に使える汎用の多目的プログラム言語。Fortran、COBOL、ALGOLなどの長所を取り入れ、計算機メーカとユーザ団体が1965年に共同で開発した。その目的は次の点にあった。

(1)あらゆる分野の進歩したデータに対しても処理が可能なこと。

(2)計算機の種類に左右されないこと。

(3)ハードウェアとオペレーティングシステムの全機能が利用できること。

PM[preventive maintenance] (同)予防保全

POL[problem oriented language] (同)問題向き言語

POS[point of sales] (関)販売管理システム

Prolog[Programming in Logic]

1970年代の初め、フランスのマルセーユ大学のA.Colmerauerらが開発したプログラム言語。1階述語論理に基礎をおいた言語。事実と規則を与えることができ、質問により答えを返すという機能をもつ。機械翻訳システムやエキスパートシステム用の言語として使う。

英字

Q

QuickTime

アップル社が開発したマルチメディア拡張ソフト。ビデオやアニメーション、音声などを再生する。あらかじめ決められた時間軸に沿って制御を行うため、使用している機種の処理能力に関係なく、画像と音声の同期をとりながら自然に再生できる。QuickTimeは、通常のビデオ映像用のVideo、再生速度を優先したビデオ映像のCinepak、画質優先のビデオ映像のComponentvideo、グラフィックスソフトで作成した動画用のAnimation、256色カラー画像用のGraphics、高品質画像のJPEGという6種類のデータ形式に対応している。

R

RAID[redundant arrays of inexpensive disk]

システムの信頼性を高めるために、ディスク装置に冗長性を導入すること、またはそのためのディスク装置。読みは「レイド」。比較的安価なディスク装置を複数台用いてアクセスを分散させることにより、高速・大容量で信頼性の高いディスク装置を実現するための技術。RAID0～RAID5までの6つのレベルがあり、RAID0でストライピング(アクセスの分散化)、RAID1でミラーリング、RAID2～5でデータとパリティ(ECC：誤り訂正)の分散書き込みを実現する。

RAM[random access memory]

CPUが読出しと書込みを直接行える記憶装置。R／Wメモリ(リード／ライトメモリ)ともいう。揮発性メモリで、電源を切断すると記憶の内容を失う。ICメモリではスタティックRAM、ダイナミックRAMなどがこれに相当する。

RAS[reliability／availability／serviceability]

信頼性(reliability)と可用性(availability)と保守容易性(serviceability)のこと。三語の頭文字を合成した語。システム利用者に対するサービス度合いの基準となっている。信頼性とは、どのような状態になってもシステムが故障しない性質。可用性とは稼働率と同じでシステムが故障しにくく、かつ故障しても復旧までの時間が短くてすむ性質。保守容易性とは、システムに障害が発生しても短時間で復旧できる性質。

これらを向上させることにより、コンピュータシステムの高信頼性と効

率のよい運用を図る基準となるものである。なお、これに完全性・安全性(integrity)と機密性(security)を加えたRASISという基準もある。

RASIS[RAS＋integrity＋security]

コンピュータシステムのハードウェア・ソフトウェア両面における信頼性の概念RASに、Integrity(完全性)とSecurity(機密性・安全性)を加え、より高い信頼性の技術を総称したもの。

RBA[relative byte address] (同)相対バイトアドレス

RDA[remote database access]

遠隔地にあるデータベースをアクセスするためのアプリケーションインタフェースのOSI標準規格。標準アクセス言語はSQL。データベースサーバ側とクライアント側のアクセス手順を障害発生時のやり取りも含めて規定している。

RDB[relational database] (同)関係データベース

RDF[record define field] (参)制御インタバル

ReadMeファイル[Read Me file]

ソフトウェアを使用する前にユーザに読んでもらいたい記事を提供者側がまとめたファイル。内容は、ソフトのインストール方法、運用方法、バージョン情報、技術情報、印刷物マニュアルの訂正などさまざまである。

ソフトの提供媒体にREADME.TXT、README.DOCなどの名称で収録されている。

RJE[remote job entry] (同)リモートジョブエントリ

RPG[Report Program Generator]

データファイルの作成、保守、検索、報告書の作成を中心とするプログラムの生成に使う表記入方式の簡易ファイル言語。各欄に記入したパラメタにより、ジェネレータが目的プログラムを生成する。報告書作成用言語として登場した。現在機能を強化し、多様な命令を追加している。

RPM、rpm[revolutions per minute]

毎分の回転数。磁気ディスク記憶装置、フロッピーディスク記憶装置等、回転の動作をともなう機器の性能表現に用いる単位。

RR[round robin] (同)ラウンドロビン

RRDS[relative record data set]

(同)相対レコードデータセット

RRN[relative record number] (同)相対レコード番号

RS-232C

ITU-T勧告V.24に基づきEIAが定めたシリアルインタフェース規格のひとつ。もともとはDTE(データ端末装置)とDCE(回線終端装置)間のインタフェースであるモデム用インタフェースとして普及した。

現在ではプリンタやディスプレイ装置などの周辺装置との標準インタフェースとして広く用いる。なお、この規格はコネクタピンの役割と電圧を定めただけである。そのため、25ピンや9ピン、ストレートケーブルやクロスケーブルといった、RS-232Cインタフェースに準拠しているにもかかわらず何種類かの非互換製品が存在する。

RSA[Rivest-Shamir-Adleman]

代表的な公開かぎ暗号化方式のひとつ。基本方式は、べき乗剰余演算を用いて暗号化・復号処理を行う。ユーザが多く、業界標準となっている。復号かぎが分かっていないと大きな数の素因数分解が必要であり、その計算時間が膨大となることを利用して暗号の安全性を保っている。しかし、コンピュータが高性能になっているので、暗号強度を高めるために1024ビット、または2048ビットのかぎを使う方向にある。3人の提唱者の名前の頭文字を並べてこの名称とした。

S

SAM[sequential access method] (関)順次アクセス

SE[systems engineer] (同)システムエンジニア

SET[secure electronic transaction]

1996年2月に米国大手クレジットカード会社のMasterCardとVISAが合意して決めた電子商取引(EC)の一連のトランザクションのセキュリティを守るための方式。それまでの電子商取引に対するセキュリティ方式は、MasterCardがネットスケープコミュニケーションズ社の技術をもとに開発したSEPP(secure electronic payment protocol)とVISAがマイクロソフトと共同で開発したSTT(secure transaction technology)の両方式に分かれていた。

SGML[Standard Generalized Markup Language]

電子文書交換のためにISOが標準化した汎用の文書定義言語。テキストに文字種、表示、印字の体裁まで含めて記述できるほか、見出し、本文、索引など文書の構造を定義できる。ISO/IEC JTC1のSC18が定め、1986年にISO規格となった文書技術言語。標準一般化マーク付け言語ともいう。SGML規約に準拠して文書を記述するとマルチメディアを意識したハイパーテキストの文書の検索、改訂、製本、版管理が容易になる。複雑な機械類の開発を委託するような場合に、SGML文書にすることにより、文書の取り扱いの容易さと正確性が増大する。そのため、CALSのようなコンピュータの支援を受けた資材取引にSGML文書の利用を検討して

いる。HTMLは、SGMLを拡張し、ページ間のリンクの指定やデータ入力フィールドを指定できる。(参)HTML、XML

Smalltalk

グラフィックディスプレイとユーザ指向のヒューマンインタフェース機能(ウィンドウ機能、マウス操作など)をサポートするオブジェクト指向言語。1970年、米国ゼロックス社のA.C.Keyらが考案した。LISPから実行機構、LOGOからグラフィックス機能、SIMURAからオブジェクト指向の概念を継承している。1980年にSmalltalk-80を出した。1982年版の仕様がほぼ標準になっている。

S/MIME[secure MIME]

MIME標準の電子メールにRSAベースの暗号化機能を付加しメールのセキュリティを高めたもの。X.509に準拠したディジタル認証管理機能ももつ。マイクロソフト、ロータス、ネットスケープ、アップル、ベリサイン、RSAデータセキュリティなどが製品やサービスの中で採用することを検討している。S/MIMEはPEM、PGPより後から出てきたものであるが、それらに比べ拡張性が高く適応範囲も広いため、今後採用製品が広がる可能性がある。(関)MIME、PEM、PGP、認証

SNA[Systems Network Architecture]

アメリカのIBM社が1974年に発表したネットワークアーキテクチャ。この発表により、ネットワークアーキテクチャという概念が一般に広まった。しかし、これは自社製品内のデータ通信の標準化であり、他社製品との接続までは考えていなかった。そのため、他のコンピュータメーカも独自のネットワークアーキテクチャを次々と発表した。この結果、どのメーカの製品でも相互に接続できる標準ネットワークアーキテクチャの必要性が重要視され、OSIの標準化へと発展していった。

SP[stack pointer] (同)スタックポインタ

SP[space]

間隔、間隔文字。(同)空白文字

SP[structured programming]

(同)構造化プログラミング

SQL[Structured Query Language]

リレーショナル構造のデータベースを定義・操作する代表的データベース言語のひとつ。SQLは、IBM社のStructured Query Languageをもとに規定され、1987年にISO規格、JIS規格になった。このとき単にSQLと称することになった。SQLは、次のデータベース言語の構文と意味を規定している。

(1)スキーマ定義言語(SQL-DDL):表のスキーマを定義する命令文の集

まり。

(2)データベース操作言語(SQL-DML):表を操作するための命令文の集まり。適用業務プログラムが使用できるほか、利用者が直接的にシステムの入出力装置を用いて利用することもできる。

SQLの検索は、SELECT〜FROM〜WHERE〜の3つの句で行い、選択、射影、結合が実現できる。SQLを採用したデータベース管理システムには、DB2(IBM社)、ORACLE(オラクル社)、SYBASE(サイベース社)、Informix(インフィミックス社)、RDBⅡ(富士通)、XMD/RD(日立製作所)などがある。(関)関係データベース

SVC[supervisor call]

実行中のプログラムが、オペレーティングシステムのサービスの提供を受けるための割込みによってスーパバイザ(監視プログラム)を呼び出すこと。

オペレーティングシステムのスーパバイザは、ユーザプログラムの実行状況を監視し、メモリ保護、I/Oやプロセッサの状態設定などを行う。このサービス機能は、誤用を防ぐ目的でスーパバイザモードだけで利用可能な特権命令とし、ユーザプログラムでの直接使用を禁じている。このためプログラムが特権命令を使用するときはSVCを行う。

(参)特権命令

SYLK[symbolic link file]

アメリカのマイクロソフト社の基本フォーマット形式のファイル。同社の表計算ソフトMultiplanやExcelのファイル形式のひとつに採用されている。SYLK形式ファイルは、他社のデータベースソフト、表計算ソフトでも利用可能。

英字

T

TAT[turnaround time]　(同)ターンアラウンドタイム

Tiff[Tagged Image File Format]

画像データを記録するファイル形式。異なる機種やアプリケーション間で画像データの互換性を実現する。単純なモノクロ画像からグレースケール、1677万色のフルカラー画像に至るまで、さまざまなグラフィックス形式に対応しており、写真などのスキャニングした画像を扱うのにも適している。タグと呼ばれる説明部分に、データをどんな形式で記録したのかを書き込まれる。

TSS[time sharing system]　(同)タイムシェアリングシステム

U

UCS[universal multiple-octet coded character set]

ISO/IEC JTC 1 SC 2が作成し、1993年に国際標準とした国際符号化文字セット。1995年にその第1部がJIS規定(X0221)となる。この規格は、世界の言語(用字)を書き表した形(表記形)および記号の表現・伝送・交換・処理・蓄積・入力・表示に適用するためののものである。なお、UCSでの漢字の扱いは、CJK(中国、日本、韓国)統合漢字となっている。

UNICODE

1文字を2バイトで表現し、アルファベット以外に漢字のような文字種の多い言語文字にも対応可能とした文字コードの国際標準。国際符号化文字集合(UCS;ISO10646/JIS　X0221)との連携が可能となっている。標準はアップル、IBM、マイクロソフトが中心となって設立したユニコードコンソーシアムが規定化。1996年に第2版が出された。漢字は、CJK統一漢字コードを使い、中国、日本、韓国で同じ文字は同一コードとなっている。また、漢字特有の異字体を別に定義したり、部首索引を提供するなどの配慮が行われている。この標準により、各国語対応したヒューマンインタフェースをもったり、インターネット経由でアルファベット以外の文字をやり取りするような多言語をサポートするソフトウェアの開発が容易になる。マイクロソフトのWindows NTの内部的な文字処理にも使用されている。(関)UCS

UNIX/unix

小型コンピュータ用のオペレーティングシステムのひとつ。ミニコンピュータからパソコンまでの共通利用や異機種間のネットワークを利用する上ですぐれた機能をもち、現在はワークステーション級のコンピュータの主流オペレーティングシステムになっている。タイムシェアリングシステム用のオペレーティングシステムで、マルチユーザマルチタスク(同時に複数の利用者が複数の仕事を行う)向けになっている。

アメリカのベル研究所のD.M.RitchieとK.Thompsonらが、1969年に開発した。ベル研究所は大学や研究機関にソースコードを公開し、それをもとに各メーカが独自OSを開発したために、各種のUNIXができた。

UPS[uninterruptible power supply system]

外部供給の電源電圧・周波数の変動が著しいところで、それらの安定化と停電時の一定時間電力供給を可能にする電源装置。フォールトトレラント装置のひとつで、無停電電源供給システムともいう。コンピュータ内の主記憶装置に記憶されているデータの消去を防ぐために、サーバのように

24時間連続で自動運転するコンピュータシステムの電源障害対策として設置する。最近は、低価格のパソコン用のものも販売されるようになり、パソコンをサーバとして利用できるようになった。

V

VDT[visual display terminal]

キーボードとCRT表示装置を備えた入出力端末。ほとんどのコンピュータ端末やネットワークに接続するワークステーションやパソコンなどが相当する。

VDT作業[visual display terminal working]

VDTを有する端末などを操作する作業のこと。マンマシンインタフェースのひとつとして、VDTは重要な地位を占める。しかし、長時間のVDT作業による健康障害問題も表面化している。

VisualBasic

マイクロソフト社が開発したWindows用のBASIC言語。GUI機能を生かしたアプリケーションを手軽に作成できるのが特徴。ワープロソフトのWordや表計算ソフトのExcelなどのマクロ言語にも用いる。視覚的プログラミングのための代表的な言語。基本となるウィンドウに、ボタンやスクロールバーなどの部品を配置し、その操作や入力されるデータに関する処理を記述することでWindows対応のアプリケーションを作成できる。ただし、他のコンパイラ型言語と違って、実行ファイルを単独で動作させることはできず、VisualBasicのランタイムモジュールを必要とする。

VMAP[volume allocation map]

ファイルの媒体空間をあらかじめ固定サイズに分割し、どのファイルがどの空間を使用するかで管理する場合の管理情報を記録する部分。大きなファイルは多数のエリアを占めることになる。VTOCと比べると記憶空間の効率を重視する方式である。次のような特徴がある。

(1)ファイルの装置独立性が高い。

(2)ファイルが拡張された場合も対応が可能である。

VOL[volume]　(同)ボリューム

VRML[Virtual Reality Modeling Language]

シリコングラフィックス社やIBMの技術者が中心となって開発した3次元グラフィックス言語。インターネットのWWWページ上に仮想空間を実現するときに用いる記述言語。WWW対応のドキュメント記述言語であるＨＴＭＬを拡張し、相互補完するものである。ＶＲＭＬのデータは、

WebSpaceという専用のWebブラウザを用いて再生できるほか、NetscapeNavigatorやInternetExplorerなどの標準的なWWWブラウザで表示するためのプラグイン(機能拡張ソフト)を用意している。

1980年代後半に欧米の研究者が、情報交換の手段としてインターネット上で3次元情報をやりとりできないかと考えたのがVRML開発のきっかけとなった。

VSAM[virtual storage access method]

仮想記憶のもとで、索引順次編成(ISAM)に相当するデータセットを直接アクセスする方法。旧来のデータ管理が提供している順アクセスの機能、直接アクセスの機能をひとつにまとめたアクセス法で、装置からの独立性が高い。VSAMでアクセスするデータセットはVSAM特有な構造をしている。VSAMデータセットに対して、順、直接あるいはスキップ順にデータの読み込み、書き出し、追加、交信、削除を行うことができる。

VSAMデータセットにはエントリ順データセット(Entry Sequenced data Set：ESDS)、キー順データセット(Key Sequenced data Set：KSDS)、相対レコードデータセット(Relative Record data Set：RRDS)がある。

VSAMファイル[virtual storage access file]

仮想記憶のもとで、索引順次編成(ISAM)に相当するデータセットを直接アクセスすることのできるファイル。記憶形式が直接アクセス装置のタイプに依存しない、記憶形式、アクセスの特徴に応じたデータセットが選択できるなどがVSAMの特徴である。

VTOC[volume table of contents]

ファイル装置内の記憶空間を、シリンダ単位、トラック単位で管理する情報を記録する部分。1個のボリュームに含まれるすべてのファイルの情報(ファイルラベル)が記録されている。ボリューム目録、ボリューム目次ともいう。VTOCを見て、ボリューム内にあるファイルの記録状況、更新状況、記録位置(トラックやシリンダなど)、記憶領域の空き状況などを把握する。VTOCの位置は、ボリュームラベル内のVTOCアドレスにより知ることができる。

VMAPに比べると実行効率を優先する方式である。次のような特徴がある。

(1) 実際の記憶媒体の構造とファイルの構造が一致し、処理目的にかなったデータ配置が可能。

(2) 管理情報はボリュームごとにそのボリュームに書き込まれるので、ボリューム単位で管理できる。

(3) ファイル管理情報は、ファイルサイズの大小であまり変動しない。頻

繁に拡張されるファイルや小容量ファイルが多い場合は効率が悪い。

W

WAN[wide area network]

コンピュータネットワークの一形態で、事業所間や企業間のコンピュータを加入電話網や公衆網を使って相互接続したネットワーク。広域ネットワーク、広域網ともいう。LANと異なり、網を提供する電気通信事業者に通信料金を支払う。また、国内の異なるWANの相互接続、別の国のWANとの相互接続といった広域網間接続も盛んになった。

Web

インターネットやイントラネット上でハイパーテキスト情報をやりとりするサーバソフトウェア。サーバ管理下のHTML記述によるハイパーテキストドキュメントを相互にリンクしたページを作成し、ブラウザからURLを指定することによって欲しいページ情報をリンクをたどり順次取り出す。WWW(World Wide Web)とも呼ぶが、イントラネット環境では必ずしもWorld WideではないためWebと読みかえる場合もある。スイスのCERN研究所が1989年に提案・開発した。その後、利用者の増大とともに製品版のWebサーバを数多く開発した。サーバの構築に比較的コンピュータの専門性を要求しないため、インターネットはこの技術によって一般に普及したと言える。Webの原義は「蜘蛛の巣」と言う意味。

WS[workstation]　(同)ワークステーション

WWW[World Wide Web]

スイスのCERNで1989年に開発された情報検索・表示システムの名称。サーバシステムのひとつで、主として情報発信に利用されており、複数のファイルを関連づけて扱える点、文字情報だけでなく音声、画像、動画も扱える点でインターネット上の代表的な情報検索システムとなった。

情報をページという単位で管理し、関連するページをリンクすることができる。リンクは世界中にあるサーバのページに行うことも可能であるため、多様な情報提供ができる。ページのレイアウト情報やリンク情報を記述するのがHTMLという言語である。(関)HTML

WYSIWIS[what you see is what I see]

グループウェアで自分と相手が同じものを見ることができること。WYSIWYGをもじって名づけたスローガン。

WYSIWYG[what you see is what you get]

ワードプロセッサや帳票作成プログラムの処理中、ディスプレイ画面上

に表示した文書や帳票を、そのままの形式でプリントアウトできる機能。「目で見たものが手に入る」というスローガンの頭文字を並べたもの。ユーザにとっては便利な機能である。

X

XML[extensible markup language]

WWWコンソーシアム(W3C)が1997年頃より本格的標準化を検討しているWeb用マークアップ言語。代表的なWeb用マークアップ言語のHTMLと同様、SGMLのサブセットである。利用者による新たな独自タグの追加機能を持つことがHTMLと大きく異なる。そのタグに対応した処理を追加することにより、より高度で目的にかなったデータ交換が可能となる。例えば、XMLを利用すると、画像、動画、音声データなどを含むオンラインカタログを利用者の環境に合わせた形で効果的に提供できるようになる。マイクロソフト、ネットスケープ、サンマイクロシステムズなどの主要なインターネットソフトベンダがXMLの製品化に力を入れている。
(関)SGML

XOR[exclusive OR] (同)EOR

XR[index register] (同)インデックスレジスタ